Arm Cortex-A520 Core (MP144)

Software Developer Errata Notice

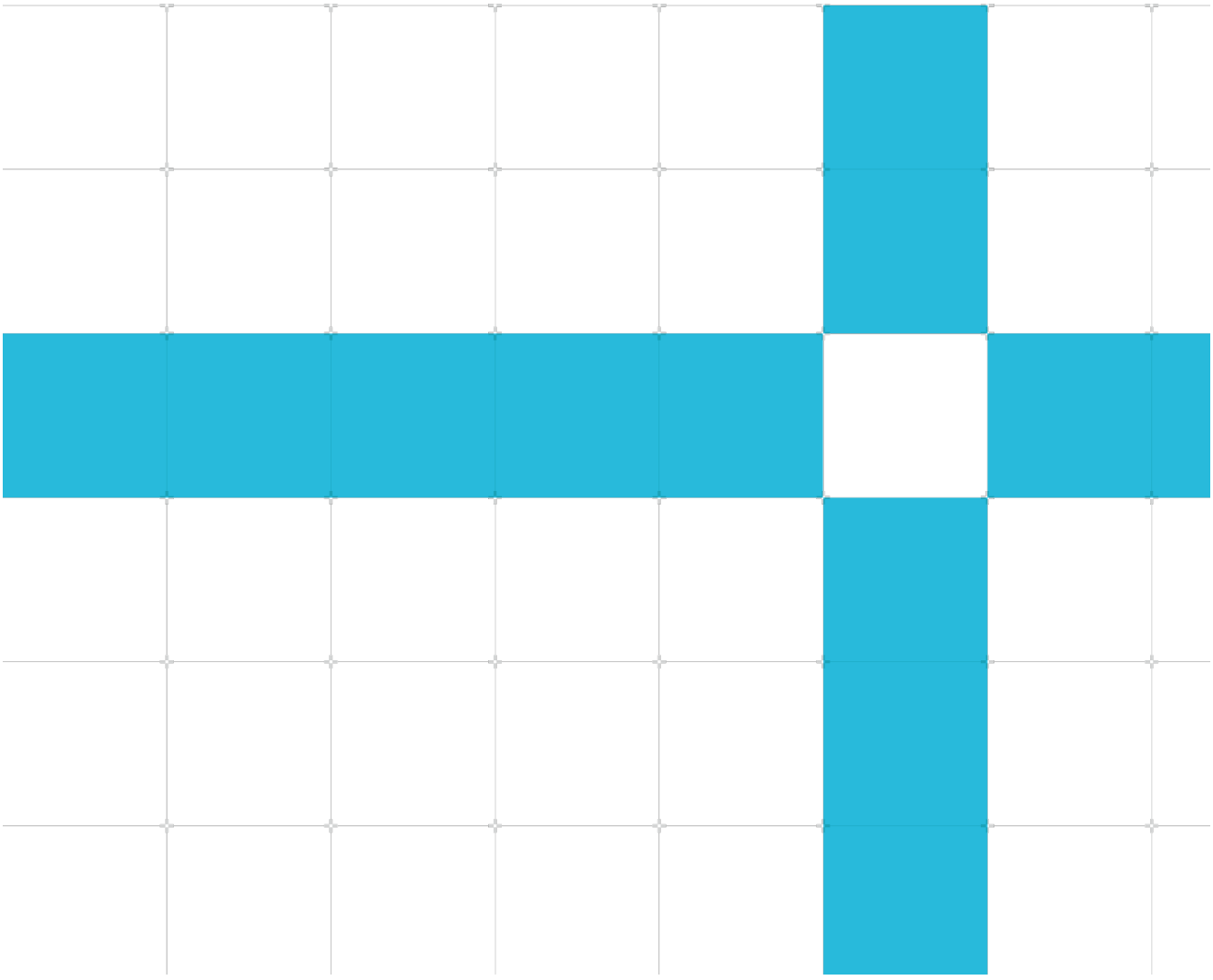
Date of issue: May 30, 2025

Non-Confidential

Document version: 10.0

Document ID: SDEN-2444153

This document contains all known errata since the r0p0 release of the product.



This document (SDEN_2444153_10.0_en) was issued on May 30, 2025.

There might be a later issue at **http://developer.arm.com/documentation/SDEN-2444153**

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

If you find offensive language in this document, please email **terms@arm.com**.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm Cortex-A520 Core (MP144), create a ticket on **https://support.developer.arm.com**.

To provide feedback on the document, fill the following survey:
https://developer.arm.com/documentation-feedback-survey.

Contents

Introduction

Scope

This document describes errata categorized by level of severity. Each description includes:

- The current status of the erratum.
- Where the implementation deviates from the specification and the conditions required for erroneous behavior to occur.
- The implications of the erratum with respect to typical applications.
- The application and limitations of a workaround where possible.

Categorization of errata

Errata are split into three levels of severity and further qualified as common or rare:

Category A	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
Category A (Rare)	A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category B	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
Category B (Rare)	A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category C	A minor error.

Change Control

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text. Fixed errata are not shown as updated unless the erratum text has changed. The **errata summary table** identifies errata that have been fixed in each product revision.

May 30, 2025: Changes in document version v10.0

ID	Status	Area	Category	Summary
3802428	New	Programmer	Category B	Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core
3975966	New	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access
3817217	New	Programmer	Category C	Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration

November 08, 2024: Changes in document version v9.0

ID	Status	Area	Category	Summary
3729900	New	Programmer	Category C	Performance might drop for core 1 in a 2-core complex configuration
3738908	New	Programmer	Category C	Consecutive CTI trigger events from the same ELA output might not be sent to the CTI
3762369	New	Programmer	Category C	Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions
3777128	New	Programmer	Category C	CTI trigger events from a core in Standby state might not be sent correctly
3777132	New	Programmer	Category C	External debug request while transitioning to emulated off might cause core to deadlock

September 09, 2024: Changes in document version v8.0

ID	Status	Area	Category	Summary
3559261	New	Programmer	Category B	Power transition to FULL_RET is denied if FUNC_RET is also enabled
3631357	New	Programmer	Category B	Store operations might modify data twice
3672344	New	Programmer	Category B	CPU non-secure physical timer event interrupt might be triggered incorrectly
3685825	New	Programmer	Category B	CAS/CASP atomic instruction might get data corruption under certain conditions
3604547	New	Programmer	Category C	Some PMU events do not count correctly
3650470	New	Programmer	Category C	Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress
3660280	New	Programmer	Category C	CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode
3680961	New	Programmer	Category C	Core might not execute some instructions in debug state after a reset catch debug event is generated
3685405	New	Programmer	Category C	The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region
3701090	New	Programmer	Category C	IRG produces biased tag generation when GCR_EL1.RRND == 1
3705320	New	Programmer	Category C	Interrupt signals generated by cores might be deasserted when in retention

May 30, 2024: Changes in document version v7.0

ID	Status	Area	Category	Summary
3189100	New	Programmer	Category B	Unmodified MTE tags might be written back to memory
2872870	Updated	Programmer	Category C	CE or DE errors from L1 data cache access might not be recorded in the RAS records
3094433	New	Programmer	Category C	The core might lose register accesses or interrupts in low-power state
3145557	New	Programmer	Category C	Some PMU events for Operation speculatively executed do not count correctly
3185964	New	Programmer	Category C	A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction
3186696	New	Programmer	Category C	The virtual address is not sign extended in EDWAR
3311443	New	Programmer	Category C	External registers PMCEID2 and PMCEID3 are not implemented
3326745	New	Programmer	Category C	Power transition from EMU_OFF to OFF might not complete
3445783	New	Programmer	Category C	Some PMU events do not count correctly
3531773	New	Programmer	Category C	TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register
3542363	New	Programmer	Category C	Load/Store instruction might get unexpected translation fault
3600964	New	Programmer	Category C	A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort

December 15, 2023: Changes in document version v6.0

ID	Status	Area	Category	Summary
2938996	Updated	Programmer	Category B	Data corruption or deadlock might happen if TRBE is enabled
2966298	Updated	Programmer	Category B	A speculatively executed unprivileged load might leak privileged data via a cache side channel
3006395	New	Programmer	Category C	PMU Events might be inaccurate
3067972	New	Programmer	Category C	AMU does not count STALL_BACKEND_MEM correctly
3103429	New	Programmer	Category C	STREX might raise both Synchronous and Asynchronous abort for external abort

September 08, 2023: Changes in document version v5.0

ID	Status	Area	Category	Summary
2738620	Updated	Programmer	Category B	Deferred error might become uncontainable
2938996	New	Programmer	Category B	Data corruption or deadlock might happen if TRBE is enabled
2966298	New	Programmer	Category B	A speculatively executed unprivileged load might leak privileged data via a cache side channel
2841875	New	Programmer	Category C	An uncontainable error might deadlock the cluster
2853709	New	Programmer	Category C	Error record registers indicate incorrect feature support in configurations without cache protection
2879977	New	Programmer	Category C	Unmodified cache line might be written back to memory
2940628	New	Programmer	Category C	Store data might be lost when a correctable error is detected in the L1 data cache
2969138	New	Programmer	Category C	Unmodified page table cache lines might be written back to memory

March 29, 2023: Changes in document version v4.0

ID	Status	Area	Category	Summary
2858100	New	Programmer	Category B	Core may deadlock due to an ECC error in L1 data cache
2487790	Updated	Programmer	Category C	ERR0MISC1 value might be incorrect after multiple simultaneous errors detected
2751027	New	Programmer	Category C	Load operation might abort unexpectedly when accessing poisoned data
2803663	New	Programmer	Category C	Speculative dirty bit hardware update might happen for store operation
2833401	New	Programmer	Category C	Direct access to internal memory might not be reliable
2861633	New	Programmer	Category C	Some PMU events are incorrectly masked to trace unit
2871911	New	Programmer	Category C	LDG or MTE checked load/store might fail to detect poisoned data
2872870	New	Programmer	Category C	CE or DE errors from L1 data cache access might not be recorded in the RAS records

December 09, 2022: Changes in document version v3.0

ID	Status	Area	Category	Summary
2630792	Updated	Programmer	Category B	Data corruption might occur during core powerdown
2738620	New	Programmer	Category B	Deferred error might become uncontainable
2710075	New	Programmer	Category C	Read value of IMP_CPUCFR_EL1 might be incorrect
2713358	New	Programmer	Category C	ERRxSTATUS.UET field might be incorrect
2713644	New	Programmer	Category C	Cache debug target for L2 Data RAM may not record correct data
2732181	New	Programmer	Category C	ERR2STATUS might be incorrect
2740664	New	Programmer	Category C	PMU event 0x77 CRYPTO_SPEC does not always count when enabled

July 29, 2022: Changes in document version v2.0

ID	Status	Area	Category	Summary
2489489	New	Programmer	Category B	Atomic instructions might use older allocation tag for tag check
2630792	New	Programmer	Category B	Data corruption might occur during core powerdown
2655133	New	Programmer	Category B	Core might deadlock during transition from ON to OFF, or ON to OFF_EMU power mode
2658327	New	Programmer	Category B	BFMMLA instructions might produce incorrect result
2677201	New	Programmer	Category B	MTE check for store might not observe correct memory ordering
2680753	New	Programmer	Category B	A core might deadlock during powerdown if TRBE is enabled
2487790	New	Programmer	Category C	ERR0MISC1 value might be incorrect after multiple simultaneous errors detected
2572702	New	Programmer	Category C	ELADISABLE does not disable APB access to the complex ELA
2604637	New	Programmer	Category C	PMU event counts might be inaccurate
2626173	New	Programmer	Category C	ERR0STATUS.SERR might be incorrect
2626511	New	Programmer	Category C	Minimum power policy might prevent power off when FUNC_RET in use
2628441	New	Programmer	Category C	External aborts might result in a deadlock
2637415	New	Programmer	Category C	Core might not execute any instruction when performing Halting Step
2640950	New	Programmer	Category C	External 32-bit writes to some 64-bit RAS registers are not mapped correctly
2668978	New	Programmer	Category C	External aborts reporting cannot be disabled
2679529	New	Programmer	Category C	Multiple simultaneous errors report for L1 data cache might be incorrect
2681778	New	Programmer	Category C	TLBI not fully invalidating entries because of parity errors
2690489	New	Programmer	Category C	Some architectural PMU events are not always available to trace unit
2708967	New	Programmer	Category C	Read value of PMMIR is incorrect

April 08, 2022: Changes in document version v1.0

ID	Status	Area	Category	Summary
2389819	New	Programmer	Category B	TLB not invalidated in complex power transitions
2441013	New	Programmer	Category B (rare)	Completion of affected memory accesses might not be guaranteed by completion of a TLBI
2567050	New	Programmer	Category C	Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock

Errata summary table

The errata associated with this product affect the product versions described in the following table.

ID	Area	Category	Summary	Found in versions	Fixed in version
2389819	Programmer	Category B	TLB not invalidated in complex power transitions	r0p0	r0p1
2489489	Programmer	Category B	Atomic instructions might use older allocation tag for tag check	r0p0	r0p1
2630792	Programmer	Category B	Data corruption might occur during core powerdown	r0p0, r0p1	r0p2
2655133	Programmer	Category B	Core might deadlock during transition from ON to OFF, or ON to OFF_EMU power mode	r0p0	r0p1
2658327	Programmer	Category B	BFMMLA instructions might produce incorrect result	r0p0	r0p1
2677201	Programmer	Category B	MTE check for store might not observe correct memory ordering	r0p0	r0p1
2680753	Programmer	Category B	A core might deadlock during powerdown if TRBE is enabled	r0p0	r0p1
2738620	Programmer	Category B	Deferred error might become uncontainable	r0p0, r0p1	r0p2
2858100	Programmer	Category B	Core may deadlock due to an ECC error in L1 data cache	r0p0, r0p1	r0p2
2938996	Programmer	Category B	Data corruption or deadlock might happen if TRBE is enabled	r0p0, r0p1	r0p2
2966298	Programmer	Category B	A speculatively executed unprivileged load might leak privileged data via a cache side channel	r0p0, r0p1	r0p2
3189100	Programmer	Category B	Unmodified MTE tags might be written back to memory	r0p2	r0p3
3559261	Programmer	Category B	Power transition to FULL_RET is denied if FUNC_RET is also enabled	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3631357	Programmer	Category B	Store operations might modify data twice	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3672344	Programmer	Category B	CPU non-secure physical timer event interrupt might be triggered incorrectly	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3685825	Programmer	Category B	CAS/CASP atomic instruction might get data corruption under certain conditions	r0p0, r0p1, r0p2, r0p3, r0p4	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3802428	Programmer	Category B	Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3975966	Programmer	Category B	Power transition might deadlock on Utility Bus or APB access	r0p0, r0p1, r0p2, r0p3, r0p4	Open
2441013	Programmer	Category B (rare)	Completion of affected memory accesses might not be guaranteed by completion of a TLBI	r0p0	r0p1
2487790	Programmer	Category C	ERR0MISC1 value might be incorrect after multiple simultaneous errors detected	r0p0, r0p1	r0p2
2567050	Programmer	Category C	Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock	r0p0	r0p1
2572702	Programmer	Category C	ELADISABLE does not disable APB access to the complex ELA	r0p0	r0p1
2604637	Programmer	Category C	PMU event counts might be inaccurate	r0p0	r0p1
2626173	Programmer	Category C	ERR0STATUS.SERR might be incorrect	r0p0	r0p1
2626511	Programmer	Category C	Minimum power policy might prevent power off when FUNC_RET in use	r0p0	r0p1
2628441	Programmer	Category C	External aborts might result in a deadlock	r0p0	r0p1
2637415	Programmer	Category C	Core might not execute any instruction when performing Halting Step	r0p0	r0p1
2640950	Programmer	Category C	External 32-bit writes to some 64-bit RAS registers are not mapped correctly	r0p0	r0p1
2668978	Programmer	Category C	External aborts reporting cannot be disabled	r0p0	r0p1
2679529	Programmer	Category C	Multiple simultaneous errors report for L1 data cache might be incorrect	r0p0	r0p1
2681778	Programmer	Category C	TLBI not fully invalidating entries because of parity errors	r0p0	r0p1
2690489	Programmer	Category C	Some architectural PMU events are not always available to trace unit	r0p0	r0p1
2708967	Programmer	Category C	Read value of PMMIR is incorrect	r0p0	r0p1
2710075	Programmer	Category C	Read value of IMP_CPUCFR_EL1 might be incorrect	r0p0, r0p1	r0p2

ID	Area	Category	Summary	Found in versions	Fixed in version
2713358	Programmer	Category C	ERRxSTATUS.UET field might be incorrect	r0p0, r0p1	r0p2
2713644	Programmer	Category C	Cache debug target for L2 Data RAM may not record correct data	r0p0, r0p1	r0p2
2732181	Programmer	Category C	ERR2STATUS might be incorrect	r0p0, r0p1	r0p2
2740664	Programmer	Category C	PMU event 0x77 CRYPTO_SPEC does not always count when enabled	r0p0, r0p1	r0p2
2751027	Programmer	Category C	Load operation might abort unexpectedly when accessing poisoned data	r0p0, r0p1	r0p2
2803663	Programmer	Category C	Speculative dirty bit hardware update might happen for store operation	r0p0, r0p1	r0p2
2833401	Programmer	Category C	Direct access to internal memory might not be reliable	r0p0, r0p1	r0p2
2841875	Programmer	Category C	An uncontainable error might deadlock the cluster	r0p0, r0p1	r0p2
2853709	Programmer	Category C	Error record registers indicate incorrect feature support in configurations without cache protection	r0p0, r0p1	r0p2
2861633	Programmer	Category C	Some PMU events are incorrectly masked to trace unit	r0p1	r0p2
2871911	Programmer	Category C	LDG or MTE checked load/store might fail to detect poisoned data	r0p0, r0p1	r0p2
2872870	Programmer	Category C	CE or DE errors from L1 data cache access might not be recorded in the RAS records	r0p0, r0p1	r0p2
2879977	Programmer	Category C	Unmodified cache line might be written back to memory	r0p0, r0p1	r0p2
2940628	Programmer	Category C	Store data might be lost when a correctable error is detected in the L1 data cache	r0p0, r0p1	r0p2
2969138	Programmer	Category C	Unmodified page table cache lines might be written back to memory	r0p0, r0p1	r0p2
3006395	Programmer	Category C	PMU Events might be inaccurate	r0p0, r0p1	r0p2
3067972	Programmer	Category C	AMU does not count STALL_BACKEND_MEM correctly	r0p0, r0p1	r0p2
3094433	Programmer	Category C	The core might lose register accesses or interrupts in low-power state	r0p0, r0p1, r0p2	r0p3

ID	Area	Category	Summary	Found in versions	Fixed in version
3103429	Programmer	Category C	STREX might raise both Synchronous and Asynchronous abort for external abort	r0p0, r0p1	r0p2
3145557	Programmer	Category C	Some PMU events for Operation speculatively executed do not count correctly	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3185964	Programmer	Category C	A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction	r0p0, r0p1, r0p2	r0p3
3186696	Programmer	Category C	The virtual address is not sign extended in EDWAR	r0p0, r0p1, r0p2	r0p3
3311443	Programmer	Category C	External registers PMCEID2 and PMCEID3 are not implemented	r0p0, r0p1, r0p2	r0p3
3326745	Programmer	Category C	Power transition from EMU_OFF to OFF might not complete	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3445783	Programmer	Category C	Some PMU events do not count correctly	r0p0, r0p1, r0p2	r0p3
3531773	Programmer	Category C	TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3542363	Programmer	Category C	Load/Store instruction might get unexpected translation fault	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3600964	Programmer	Category C	A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3604547	Programmer	Category C	Some PMU events do not count correctly	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3650470	Programmer	Category C	Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3660280	Programmer	Category C	CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3680961	Programmer	Category C	Core might not execute some instructions in debug state after a reset catch debug event is generated	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3685405	Programmer	Category C	The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region	r0p0, r0p1, r0p2, r0p3, r0p4	Open

ID	Area	Category	Summary	Found in versions	Fixed in version
3701090	Programmer	Category C	IRG produces biased tag generation when GCR_EL1.RRND == 1	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3705320	Programmer	Category C	Interrupt signals generated by cores might be deasserted when in retention	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3729900	Programmer	Category C	Performance might drop for core 1 in a 2-core complex configuration	r0p0, r0p1, r0p2, r0p3	r0p4
3738908	Programmer	Category C	Consecutive CTI trigger events from the same ELA output might not be sent to the CTI	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3762369	Programmer	Category C	Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3777128	Programmer	Category C	CTI trigger events from a core in Standby state might not be sent correctly	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3777132	Programmer	Category C	External debug request while transitioning to emulated off might cause core to deadlock	r0p0, r0p1, r0p2, r0p3, r0p4	Open
3817217	Programmer	Category C	Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration	r0p0, r0p1, r0p2, r0p3, r0p4	Open

Errata descriptions

Category A

There are no errata in this category.

Category A (rare)

There are no errata in this category.

Category B

2389819
TLB not invalidated in complex power transitions

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a core that is part of a complex powers on at the same time that the other core in the same complex is powering off, then the L2 *Translation Lookaside Buffer* (TLB) in the complex might not get invalidated correctly.

Configurations Affected

This erratum only affects configurations with two cores in a complex.

Conditions

1. One core in a complex is in the OFF power mode.
2. The other core in the same complex makes a transition from ON to OFF or OFF_EMU.
3. At the time the second core has almost completed its power transition, the first core starts a transition from OFF to ON.
4. A third core, outside the complex, executes a TLB invalidate instruction that would invalidate an entry that is currently held in the L2 TLB in the complex.

Implications

The L2 TLB in the complex is not invalidated by the power sequences, and so the core retains its previous state. However, there is a brief window in which the complex is disconnected from coherency with the rest of the system and so any TLB invalidate DVMs received during this time will not take effect. This can leave stale entries in the TLB that the cores in the complex might hit when they start executing code again.

Workaround

After a core is powered ON, the firmware should execute one of the following sequences before enabling the MMU, depending on whether the system expects to use Secure EL2 or not.

Without Secure EL2:

```
TLBI ALLE3
Set SCR_EL3.NS=0
ISB
TLBI ALLE1
Set SCR_EL3.NS=1
ISB
TLBI ALLE2
TLBI ALLE1
DSB SY
```

With Secure EL2:

```
TLBI ALLE3
Set SCR_EL3.NS=0
Set SCR_EL3.EEL2=1
ISB
TLBI ALLE2
TLBI ALLE1
Set SCR_EL3.NS=1
ISB
TLBI ALLE2
TLBI ALLE1
DSB SY
```

2489489
Atomic instructions might use older allocation tag for tag check

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

An atomic instruction might use an allocation tag that is no longer current for its tag check if relying on acquire/release-based ordering.

Configurations Affected

This erratum affects all configurations where the **BROADCASTMTE** pin is HIGH.

Conditions

The erratum occurs if the following conditions are met:

1. MTE checking is enabled.
2. The core executes an instruction with acquire semantics, other than an atomic instruction.
3. The core executes a tag-checked atomic instruction within 5 instructions of the instruction above, with no intervening `DMB`.
4. Timing-sensitive, microarchitectural conditions occur.

Implications

If the conditions are met, the tag read of the tag-checked atomic instruction might not be correctly ordered w.r.t. the preceding instruction with acquire semantics. This can result in a tag check using an allocation tag for the tag check that is no longer current, if another Processing Element has modified the allocation tag concurrently.

Workaround

This erratum can be avoided by the following instruction sequence:

```
MOVZ X1, #0x0000, LSL #0
MSR S3_6_C15_C4_0, X1

MOVK X1, #0x0000, LSL #0
MOVK X1, #0x0380, LSL #16
MOVK X1, #0x0000, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_2, X1
```

```
MOVK X1, #0x0000, LSL #0
MOVK X1, #0x1FE0, LSL #16
MOVK X1, #0x0008, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_3, X1

MOVK X1, #0x03F1, LSL #0
MOVK X1, #0x0110, LSL #16
MOVK X1, #0x0000, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_1, X1

ISB
```

This is not expected to have a material performance impact in common use cases.

2630792
Data corruption might occur during core powerdown

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Data corruption might occur when a core is powered down.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core executes a PRFM PLDL1 or PRFM PSTL1 shortly before a WFI.
2. The core is requested to power off.
3. Timing sensitive microarchitectural conditions occur.

Implications

If the conditions are met, dirty data might be lost on the cache line accessed by the PRFM instructions, resulting in data corruption.

Workaround

To prevent this erratum from occurring, software can set IMP_CPUACTLR_EL1[38] = 1, for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #38, #1
MSR S3_0_C15_C1_0, x0
```

This might impact the effectiveness of some PRFM instructions. This is unlikely to have a measurable performance impact.

2655133
Core might deadlock during transition from ON to OFF, or ON to OFF_EMU power mode

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The core might deadlock during a transition from ON to OFF, or ON to OFF_EMU power mode.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. MMU is on
2. Hardware dirty bit update is enabled by SCTLR_ELx.HD or VTCR_EL2.HD
3. Core is transitioning from ON to OFF, or ON to OFF_EMU power mode
4. Timing sensitive microarchitectural conditions occur

Implications

If the previous conditions are met, the core might deadlock.

Workaround

To prevent this erratum from occurring during a transition from ON to OFF or ON to OFF_EMU power mode, the software must insert an ISB instruction before the WFI instruction.

2658327
BFMMLA instructions might produce incorrect result

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

When both cores in a complex are executing chained multiply-accumulate instructions, then under precise timing conditions a **`BFMMLA`** instruction might produce an incorrect result.

Configurations Affected

This erratum affects any configuration with 2 cores in a complex, sharing a 2x64-bit VPU datapath (configuration parameter VPU_DATAPATH is set to 2x64). Configurations sharing a 2x128-bit datapath are unaffected. Affected configurations can be identified by reading IMP_CPUCFR_EL1 using **`MRS Xt, S3_0_C15_C0_0`** - the core is affected if bit [16] is 1 and bit [4] is 0.

Conditions

This erratum occurs under the following conditions:

1. Both cores in the complex are executing chained multiply-accumulate instructions.
2. One core in the complex flushes a chained multiply-accumulate instruction executed speculatively.
3. A **`BFMMLA`** or multicycle vector instruction is executed after the flush.
4. Precise microarchitectural timing conditions occur.

Any of the following instructions are classed as chained multiply-accumulate instructions:

- **`BFMMLA`**
- **`BFDOT`**

Any of the following instructions are classed as multicycle vector instructions:

- **`FDIV*`**
- **`FSQRT`**
- **`SDIV*/UDIV*`**
- **`BDEP/BEXT/BGRP`**
- **`PMULL*`**
- **`RAX1`**
- **`SHA512*`**

- **SM3*/SM4E***

Implications

If these conditions are met, the result of a **BFMMLA** executed by either core might be incorrect. As FEAT_BF16 is a recent addition to the architecture, these instructions are not expected to be present in legacy code.

Workaround

There is no complete workaround for this erratum. It is expected that software will use run-time feature detection to determine whether to use these instructions or to fall back on support for earlier architecture versions. A kernel can avoid this erratum by updating the detected feature list to remove FEAT_BF16 from the list of supported features in affected systems.

2677201
MTE check for store might not observe correct memory ordering

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Memory Tagging Extension (MTE) check on a store might not be correctly ordered with respect to an earlier DMB or instruction with acquire semantics.

Configurations Affected

This erratum affects configurations with BROADCASTMTE=true.

Conditions

This erratum occurs under the following conditions:

1. MTE checking is enabled, by setting (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn != 0b00).
2. The core executes an instruction with acquire semantics, or a DMBLD/DMBSY.
3. A checked store instruction is executed to Inner Writeback and Outer Writeback, tagged, memory.
4. Timing-sensitive, micro-architecural conditions occur.

Implications

If the conditions are met, the MTE check on the store might not be correctly ordered w.r.t. the DMB or instruction with acquire semantics. This might result in an incorrect update of the TFSR_ELx register, or MTE check abort wrongly generated or missed to generate.

Workaround

To prevent this erratum from occurring for DMB operation, software can set IMP_CPUACTLR_EL1[10] = 1, for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #10, #1
MSR S3_0_C15_C1_0, x0
```

To prevent this erratum from occurring for Load Acquire operation, software can use the following instruction sequence

```
MOVZ X1, #0x0000, LSL #1
MSR S3_6_C15_C4_0, X1

MOVK X1, #0x0000, LSL #0
MOVK X1, #0x0850, LSL #16
MOVK X1, #0x0000, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_2, X1

MOVK X1, #0x0000, LSL #0
MOVK X1, #0x1F70, LSL #16
MOVK X1, #0x0008, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_3, X1

MOVK X1, #0x03F1, LSL #0
MOVK X1, #0x00C0, LSL #16
MOVK X1, #0x0000, LSL #32
MOVK X1, #0x0000, LSL #48
MSR S3_6_C15_C4_1, X1
ISB
```

2680753
A core might deadlock during powerdown if TRBE is enabled

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0. Fixed in r0p1.

Description

A core might deadlock during powerdown if *TRace Buffer Extension* (TRBE) is enabled.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The TRBE is enabled by setting TRBLIMITR_EL1.E = 0b1. The core is executing in a non-prohibited trace region.
2. The core executes a WFI, WFIT, WFE or WFET instuction.
3. An external wakeup or timeout occurs, waking up the core.
4. The core executes a WFI instruction to power down the core.
5. system rely on the core to be ready to enter to power OFF/OFF_EMU state
6. Timing-sensitive micro-architectural conditions occur.

Implications

If the conditions are met, the core might not accept the power state transition to OFF/OFF_EMU state and system may deadlock.

Workaround

Software can execute a **`TSB CSYNC`** and **`DSB before execute WFI for power down`**.

2738620
Deferred error might become uncontainable

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Poison information cached in a *Processing Element* (PE) might be lost and therefore make the deferred error to become uncontainable.

Configurations affected

This erratum affects configurations having parameter CORE_CACHE_PROTECTION set to TRUE and any of the following:

1. The parameter BROADCASTMTE is set to TRUE.
2. The interconnect does not have a precise snoop filter, and does not use SnpQuery to inquire about the state of the line at the *Request Node* (RN).

Conditions

This erratum might occur if line A (Normal Inner Write-Back, Outer Write-Back Cacheable) is cached in the PE and another cache in the system, one cache has the line as poisoned, and one of the following conditions is met:

1. A MakeReadUnique from the complex is processed by the interconnect and poisoned data is returned, without the line being lost by the complex.
2. The non-L1 allocating store operation executed by this PE and the store operation modifies less than a cache line worth of MTE tags.

Implications

If the condition occurs, the line might be propagated to the core without being poisoned.

Workaround

Software can enable the Error Recovery Interrupt for deferred error by setting the DUI bit of all the *Reliability, Availability, and Serviceability* (RAS) node registers ERRxCTLR, and treat all deferred errors as uncontainable.

2858100
Core may deadlock due to an ECC error in L1 data cache

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

A deadlock might occur as a result of a load accessing a L1 data cache line which has ECC error.

Configurations Affected

This erratum affects configurations with CPU_CACHE_PROTECTION set.

Conditions

The erratum occurs under the following conditions:

1. A load or store sees an ECC error or deferred error in the L1 data cache.
2. Very unlikely, timing sensitive microarchitectural conditions occur.

Implications

If the above conditions are met, the core might deadlock. The core will continue normal operation if any coherency operation is seen.

Workaround

There is still substantial benefit being gained from the ECC logic. There might be a small increase in overall system failure rate due to this erratum.
To prevent this erratum from occurring, software can set IMP_CPUACTLR_EL1[29] = 1, for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #29, #1
MSR S3_0_C15_C1_0, x0
```

This will reduce the effectiveness of internal clock gating, and might impact power efficiency. During power testing of sample silicon, Arm recommends not applying the workaround.

2938996
Data corruption or deadlock might happen if TRBE is enabled

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Data might be corrupted if *TRace Buffer Extension* (TRBE) is enabled.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The TRBE is enabled by setting TRBLIMITR_EL1.E = 0b1.
2. The trace data store to the memory cross 4k page.
3. One of the 4k page has translation fault or permission fault.
4. Timing-sensitive micro-architectural conditions occur.

Implications

If the above conditions are met, random data might be corrupted or deadlock might happen before the core takes the TRBE IRQ.

Workaround

The EL3 firmware can prevent trace collection via TRBE by programming MDCR_EL3.NSTB[1] to the opposite value of SCR_EL3.NS on a security state switch.

2966298
A speculatively executed unprivileged load might leak privileged data via a cache side channel

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

A speculatively executed unprivileged load might leak privileged data via a cache side channel.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. A load is speculatively executed at EL0, accessing a location in memory that is mapped, but lacks EL0 access permissions.
2. Timing-sensitive, microarchitectural conditions occur.

Implications

The speculatively executed load can, under specific microarchitectural conditions, speculatively forward data, bypassing a permission check, to the address operand of another load, potentially leaking information from a higher privilege level via side channel.

Pagetable isolation between EL0 and higher level ELs prevents the issue from occurring.

Workaround

If pagetable isolation is disabled, the context switch logic in the kernel can be updated to execute the following sequence on affected cores before exiting to EL0, and after all explicit memory accesses:

1. A non-shareable TLBI to any context and/or address, including unused contexts or addresses, such as a `TLBI VALE1 Xzr`.
2. A DSB NSH to guarantee completion of the TLBI.

3189100
Unmodified MTE tags might be written back to memory

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p2. Fixed in r0p3.

Description

MTE tags for a location might be marked as dirty even without a STG* instruction modifying them, which might result in those MTE tags being written back to memory.

Configurations affected

This erratum affects platforms supporting MTE (parameter BROADCASTMTE = true) that try to re-use the 3% of MTE RAM as both tags and regular memory during the runtime of the system.

Conditions

This erratum occurs under the following conditions:

1. Memory location A is marked as Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. The core allocates location A into the L1 data cache or the L2 cache in Unique state with MTE tags. This allocation might be due to committed instructions, speculative execution, or data prefetching.
3. The core executes a store operation that does not modify MTE tags.
4. Another PE is requesting line A, or line A is naturally evicted. The core will provide the MTE tags for line A marked as dirty, but their value remains unchanged.

Implications

If the previous conditions are met, the MTE tags for memory location A might be marked as modified, and subsequently be written back, replacing the original MTE tag value.
When MTE tag memory is only used for tags and no external modification occurs, this is harmless.
If the Operating System re-uses the MTE tag memory as regular memory the tag write back might occur once the memory is in use as regular memory, causing data corruption.

Workaround

If the interconnect always fetches tags, regardless of the CPU memory attributes:

- No re-use of the MTE tag memory is possible. This memory should not be described in firmware tables.

If the interconnect only fetches tags for memory that is marked as tagged, when changing the use of a page of memory from MTE tags to regular memory, the OS should:

- Remove all tagged mappings for the target page (data page corresponding to the tag page being re-used) to prevent fetching of data as tags.
- Clean and Invalidate the target page, via DC IGVAC, to remove any tags that have been cached.

3559261
Power transition to FULL_RET is denied if FUNC_RET is also enabled

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A power transition to FULL_RET will be incorrectly denied if FUNC_RET is also enabled.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core has non-zero bits in the IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL or WFE_RET_CTRL fields, enabling the FULL_RET power mode.
2. The core has non-zero bits in the IMP_CPUPWRCTLR_EL1.VPU_PWR_CTRL field, enabling the FUNC_RET power mode.
3. The core PPU requests a transition to FULL_RET.

Implications

If the previous conditions occur, the power transition to FULL_RET will be denied, and the core will not enter into FULL_RET power mode. This means that only one of the two power modes can be in use at any time, limiting the amount of power savings that are achievable.

Workaround

These modes are disabled by default. Firmware should enable only FUNC_RET or FULL_RET, but not both. The choice of mode will depend on whether FUNC_RET or FULL_RET gives greater power savings, which will depend on the implementation and expected workloads.

3631357
Store operations might modify data twice

Status

Fault Type: Programmer Category B
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

Non-L1 allocating store instructions of less than four Memory Tagging Extension (MTE) tags for the same cacheline, with full cacheline data update to the same cacheline, might modify data twice.

Configurations affected

This erratum affects configurations with two cores in a complex.

Conditions

The erratum occurs under the following conditions:

1. Cacheline X is cached in the complex in Unique state, without MTE tags.
2. Processing Element (PE) A in the same complex executes a number of store instructions that collectively modify all the data of cacheline X but not all MTE tags for X.
3. The stores are gathered and do not allocate in the L1 cache.
4. PE B outside the complex does a store to cacheline X.
5. Unlikely timing and micro-architecture conditions occur.

Implications

If the conditions are met, a PE observing X might see the cacheline

- first with the value written by PE A
- then with the value written by PE B
- again with the value written by PE A

Workaround

Write streaming can be disabled for MTE stores by setting IMP_CPUACTLR_EL1.MTEALLCWSDIS to 0b1, for example using the following code sequence

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #27 #1
MSR S3_0_C15_C1_0, x0
```

3672344
CPU non-secure physical timer event interrupt might be triggered incorrectly

Status

Fault type: Programmer Category B
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

The **nCNTPNSIRQ** interrupt might be triggered at an incorrect time when the core is in FULL_RET mode.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. EL2Enabled() i.e. SCR_EL3.{NS, EEL2} is not {0, 0}.
2. SCR_EL3.ECVEn is 1,
3. CNTHCTL_EL2.ECV is 1.
4. HCR_EL2.{E2H, TGE} is not {1, 1}
5. CNTP_CTL_EL0.{IMASK, ENABLE} is set {0,1}
6. The core executes a WFI, WFIT, WFE, or WFET instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then

- CNTPOFF_EL2 is ignored and is not applied to the PhysicalCountInt() to derive virtual counter value that is used to trigger nCNTPNSIRQ interrupt.
- **nCNTPNSIRQ** interrupt might be triggered at an incorrect time.

Workaround

Disable entering full retention mode by setting both IMP_CPUPWRCTLR_EL1.{WFE_RET_CTRL} and IMP_CPUPWRCTLR_EL1.{WFI_RET_CTRL} to 3'b000.

3685825
CAS/CASP atomic instruction might get data corruption under certain conditions

Status

Fault type: Programmer Category B
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A CAS/CASP atomic instruction executed as near atomic instruction might get silent data corruption under certain conditions.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core executes an atomic CAS/CASP instruction.
2. The previously mentioned atomic instruction hits against an ongoing L1 linefill to the same cache line as the CAS/CASP instruction.
3. Unlikely timing and micro-architectural conditions occur.

Implications

If the previous conditions are met, the atomic CAS/CASP might never perform the write even in case of successful comparison, causing the CAS/CASP data to be lost.

Workaround

To prevent this erratum from occurring, software can set IMP_CPUACTLR_EL1[9] = 1, for example using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #9, #1
MSR S3_0_C15_C1_0, x0
```

3802428
Complex power transition might deadlock when a Utility bus or Debug APB access is sent to a core

Status

Fault type: Programmer Category B
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A power transition or warm reset on both cores in a complex might deadlock when there is an access on the Utility bus or Debug Advanced Peripheral Bus (APB) interface during the transitions.

Configurations affected

This erratum affects all configurations with two cores in a complex.

Conditions

The erratum occurs under the following conditions:

- One core in the complex is transitioning from the OFF or OFF_EMU power mode to ON.
- The other core in the complex is either performing a warm reset requested by the Reset Management Register (RMR_ELx) or is also transitioning from the OFF or OFF_EMU power mode to ON.
- An access is made on the Utility bus or Debug APB interface to a memory mapped register in one of the cores while the power transitions are in progress.

Implications

When the previous conditions occur, the complex might deadlock. The power transition will not complete, and the Utility bus or Debug APB transaction will not complete.

Workaround

When the system cannot prevent Utility bus and Debug APB accesses to cores that have not yet reached the ON power mode, then it must restrict the power transitions that can occur on the two cores in parallel.

- In case of multiple cores power transitioning to ON at the same time, firmware can control the power transitions using the LOCK feature (PPU_PWPR.LOCK_EN) in the core's Power Policy Unit (PPU) such that only one core of the complex can transition to ON at a time.

- When EL3 firmware must make use of RMR_EL3, it must coordinate with an external agent to ensure a core reset due to RMR_EL3.RR does not occur at the same time as a power transition for another core in the complex.

3975966
Power transition might deadlock on Utility Bus or APB access

Status

Fault type: Programmer Category B
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A power transition might deadlock if there is an access to a core register on the Utility Bus or debug APB interface during the transition.

Configurations affected

This erratum affects all configurations of the core. However it is only realistic to hit all the conditions on configurations with no L2 cache.

Conditions

The erratum occurs under the following conditions:

1. The core power transitions from ON to OFF, or OFF_EMU to OFF.
2. An access is made on the Utility Bus or debug APB interface to access a memory mapped register in the core during the ON to OFF power transition, or an access is made on the debug APB interface to access a memory mapped register in the core during the OFF_EMU to OFF power transition. For Utility bus accesses, the address must be in the range 0x<n>9_0000 to 0x<n>F_FFFF where <n> is the core number.
3. The Utility Bus or debug APB access must start before the complex power transition starts, but because of delays on the internal buses of the cluster, the transaction does not complete until after the end of the complex power transition. The complex power transition includes flushing of the L2 cache which will take many hundreds of cycles. It is implausible for the transaction to be delayed this long, therefore this condition is only realistic on a configuration that has no L2 cache.

Implications

If the previous conditions occur, the Utility Bus or APB access might not complete, leading to a system deadlock.

Workaround

There is a workaround that avoids the problem but might not be possible to apply in many systems. There is a second partial workaround that does not cover all cases, but can be used if the first workaround is not suitable. Only one of these two options should be applied:

1. If the system and software can ensure that no Utility Bus or APB accesses can be made during a powerdown sequence, then this will avoid the erratum.
2. The system component that is programming the Power Policy Units (PPUs), typically a System Control Processor (SCP), should ensure that any core power transition from ON to OFF is replaced by the following sequence: ON to OFF_EMU to OFF. This will prevent Utility Bus accesses from causing the issue, but will not prevent APB accesses from causing a problem, therefore the problem can still occur during debug.
   If the PPUs are being used in static mode, then the SCP that is requesting the transition can request the additional transition in the sequence.
   If the PPUs are being used in dynamic mode, then the following sequence will ensure that all transitions to OFF power mode are made using the OFF_EMU mode:
   - Set the PPU_PWPR.LOCK_EN bit if it is not already set.
   - Set PPU_PWPR.PWR_POLICY to OFF_EMU instead of OFF.
   - When the PPU reaches the OFF_EMU mode, the PPU_PWPR.PWR_POLICY field should be reprogrammed to OFF. Either of the LOCKED_IRQ_MASK or EMU_ACCEPT_IRQ_MASK bits in the PPU_IMR register can be cleared to request an IRQ to be generated so that the SCP knows when this mode is reached.
   - The PPU_PWCR.PWR_DEVACTIVEEN field should be cleared to ensure that a new wakeup event arriving does not prevent the transition to OFF.
   - The SCP should wait for the PPU to reach OFF.
   - Once the PPU has reached OFF, the PPU_PWCR.PWR_DEVACTIVEEN field can be restored to its previous value.
   - The PPU_PWPR.PWR_POLICY should be set back to OFF_EMU. This can be done either as soon as the PPU has reached OFF, or it can wait until immediately before the PPU_UNLK register is written when the core is requested to power on again.

Category B (rare)

2441013
Completion of affected memory accesses might not be guaranteed by completion of a TLBI

Status

Fault Type: Programmer Category B (rare)
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The core might not guarantee completion of all memory accesses after completion of a TLB Invalidate (`TLBI`) instruction affecting those accesses on another core.

Configurations affected

This erratum affects all configurations.

Conditions

1. Another PE in the system executes a `TLBI` or Instruction Cache (`IC`) instruction, followed by a Data Synchronization Barrier (`DSB`) instruction.
2. The core executes a store to a memory location A.
3. Another PE in the system modifies the descriptor used by the store to memory location A, using a break-before-make sequence.
    - The break-before-make sequence will include a `TLBI` instruction, followed by a `DSB` instruction.
4. Rare, timing-sensitive, microarchitectural conditions occur.

Implications

The `DSB` used after the `TLBI` as part of the break-before-make sequence might not guarantee the completion of the store to memory location A under very rare and unlikely timing conditions. For most systems and applications, the latency of the break-before-make sequence and time until later reuse is very likely to exceed the latency required to naturally complete the store.

Workaround

Given the rarity of the conditions needed to trigger this erratum, a workaround is not expected to be needed in most systems.
If a workaround is required, then the `TLBI`, `DSB` sequence from the break-before-make sequence can be repeated. After repeating the `TLBI`, `DSB` sequence, all memory accesses that use a translation changed by the break-before-make sequence will have completed.

Category C

2487790
ERR0MISC1 value might be incorrect after multiple simultaneous errors detected

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

If multiple *Error Correcting Code* (ECC) errors are simultaneously detected in the L1 data cache, the reported ERR0MISC1.Bank value and/or ERR0MISC1.Granule field value for the L1 data cache MTE data RAMs might be incorrect.

Configurations Affected

This erratum affects configurations where the CORE_CACHE_PROTECTION parameter is TRUE.

Conditions

This erratum occurs under the following conditions:

1. ECC errors are detected in multiple banks in the same cycle in either the L1 data cache data RAMs or the L1 data cache MTE data RAMs.
2. At least one of the errors is of a higher severity than one of the others.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions are met, the value reported in the ERR0MISC1.Bank and/or ERR0MISC1.Granule field in the core RAS register block might be incorrect, and report the RAM bank/granule of the error in the lowest-numbered bank/granule rather than the bank/granule of the highest-priority error. The Array, Entry, SubBank fields will be correct (these field values are the same for all the simultaneously detected errors). The ERR0STATUS and ERR0MISC0 registers will correctly report the highest-priority error, and the ERR0MISC1.Granule is still correct for L1 data cache data RAM.

Workaround

No workaround is required.

2567050
Double-bit errors in the duplicate L1 data cache tag RAMs might lead to loss of coherency or deadlock

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If a double-bit error is detected on a read of the duplicate L1 data cache tag RAMs, a deadlock might occur.

Configurations Affected

This erratum affects configurations with core cache protection enabled.

Conditions

The erratum can occur under two sets of conditions.

Set 1:

1. A `DC ISW`, `DC CSW`, or `DC CISW` operation to the L1 data cache is executed. No error is detected on the read of the duplicate L1 data cache tag RAMs.
2. The same core performs an L1 data cache refill for a line in the same set as the set/way operation above. This refill might be speculative.
3. A double-bit error occurs on a read of the duplicate L1 data cache tag RAMs for the L1 cache refill above.

Set 2:

1. A new or deferred error is detected in the L1 data cache tag, data, dirty, or Memory Tagging Extension (MTE) RAMs. This causes a hardware set/way operation being generated.
2. For the hardware-initiated set/way maintenance operation above, no error is detected on the read of the duplicate L1 data cache tag RAMs.
3. The same core performs an L1 data cache refill for a line in the same set as the set/way operation above. This refill might be speculative.
4. A double-bit error occurs on a read of the duplicate L1 data cache tag RAMs for the L1 data cache refill above.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum. If the conditions are met, coherency might be lost or a deadlock might occur on later memory accesses. Double-bit errors on the duplicate L1 data cache tag RAMs are classed as uncontainable, and Arm believes these implications are in line with those of an uncontainable error. The detection and reporting of the error are unaffected.

Workaround

The first set of conditions can be avoided by software. The use of data cache maintenance by set/way operations to the L1 data cache is not necessary as the core performs automatic cache maintenance on powerup and powerdown. Where software intends to use set/way operations regardless, the data cache should be turned off to ensure the intended behavior of cleaning the cache and avoiding a speculative refill to the cache during the sequence.

The second set of conditions does not have a workaround. However, a double-bit error occurring in the duplicate tag RAMs on a cache line that has previously had another error in the L1 data cache is very unlikely.

2572702
ELADISABLE does not disable APB access to the complex ELA

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

ELADISABLE is a cluster configuration signal. When it is HIGH, it disables access to all ELAs in the cluster. Due to this erratum, if an ELA is configured in the complex, it is still accessible through the debug APB interface when ELADISABLE is HIGH. The ROM table entry for the complex ELA will still be removed, so it is not discoverable by an external debugger.

Configurations Affected

All configurations with the complex ELA are affected.

Conditions

The erratum occurs if all the following conditions apply:

1. ELADISABLE is HIGH
2. A Debug APB access is made to the memory-mapped region for the complex ELA

Implications

This erratum means that the complex ELA cannot be completely disabled. Assuming the correct address offset is already known, a debugger will have free control of the trigger and trace features. The ELA can also stop the core clock, which is its only invasive debug feature. This can still be disabled through the authentication interface (DBGEN and SPIDEN), but this also disables all other invasive debug features in the cluster.

Workaround

This erratum has no workaround.

2604637
PMU event counts might be inaccurate

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The following performance events might be unreliable due to this erratum:

- 0x811D BR_IND_RETIRED
- 0x8116 BR_INDNR_PRED_RETIRED
- 0x8117 BR_INDNR_MIS_PRED_RETIRED
- 0x810C BR_INDNR_TAKEN_RETIRED
- 0x8110 BR_IMMED_PRED_RETIRED
- 0x8111 BR_IMMED_MIS_PRED_RETIRED
- 0x811C BR_PRED_RETIRED
- 0x8114 BR_RETURN_PRED_RETIRED
- 0x8115 BR_RETURN_MIS_PRED_RETIRED
- 0x0019 BUS_ACCESS
- 0x0061 BUS_ACCESS_WR
- 0x818D BUS_REQ_RD
- 0x8125 BUS_REQ_RD_PERCYC
- 0x8128 DTLB_WALK_PERCYC
- 0x8129 ITLB_WALK_PERCYC
- 0x0040 L1D_CACHE_RD
- 0x0041 L1D_CACHE_WR
- 0x00C4 L1D_WS_MODE_ENTRY
- 0x0016 L2D_CACHE
- 0x0020 L2D_CACHE_ALLOCATE
- 0x8155 L2D_CACHE_HWPRF
- 0x81BD L2D_CACHE_REFILL_HWPRF
- 0x0018 L2D_CACHE_WB
- 0x0051 L2D_CACHE_WR
- 0x002B L3D_CACHE
- 0x400B L3D_CACHE_LMISS_RD
- 0x4021 LD_ALIGN_LAT
- 0x0072 LDST_SPEC
- 0x4020 LDST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_RD_CHECKED
- 0x4026 MEM_ACCESS_WR_CHECKED
- 0x8121 MEM_ACCESS_RD_PERCYC
- 0x4022 ST_ALIGN_LAT

- 0x0024 STALL_BACKEND
- 0x8165 STALL_BACKEND_L1D

Configurations Affected

- For the STALL_BACKEND_L1D event, this erratum affects all configurations with L2 cache.
- For the other events, this erratum affects all configurations.

Conditions

No specific conditions are needed.

Implications

The affected events have been divided into categories:

- Events in the Low impact category still produce an indicative result in most cases.
- Events in the High impact category are too inaccurate to be used, unless they have a workaround.

Low impact:

- The L2D_CACHE_WB, L2D_CACHE, L2D_CACHE_HWPRF and L2D_CACHE_REFILL_HWPRF events might overcount if a request sees a protocol-level retry. Protocol-level retries between the core and the DSU are not common.
- The STALL_BACKEND event might slightly undercount. The STALL_BACKEND_L1D event might slightly overcount.
- The L3D_CACHE, L2D_CACHE_WR, and L2D_CACHE_ALLOCATE events might undercount.
- The LDST_SPEC event might undercount. The sum of LD_SPEC and ST_SPEC events gives the expected result.
- Long latency misses might cause L1D_CACHE_RD, L1D_CACHE_WR, and L1D_WS_MODE_ENTRY events to overcount.
- Long latency misses might cause STALL_BACKEND_L2D event to undercount.

High impact, with complete workaround:

- The affected BR events can overcount or undercount significantly depending on whether other BR events are enabled or not. The affected events will count accurately if all of the following are also enabled: PC_WRITE_RETIRED, BR_RETIRED, BR_IMMED_RETIRED, BR_RETURN_RETIRED, BR_MIS_PRED_RETIRED.
- LD_ALIGN_LAT, ST_ALIGN_LAT, LDST_ALIGN_LAT, and the affected MEM_ACCESS* events will not count at all unless an event number in the range 0 to 0x1FF is also enabled. Otherwise, they are accurate.

- L3D_CACHE_LMISS_RD might undercount significantly. 0x00A2 L3D_CACHE_REFILL_RD can be used instead as a workaround.

High impact, with approximate or no workaround:

- The BUS_REQ_RD event might significantly overcount. If the number of Inner-Writeback and Outer-Writeback cacheable accesses far exceeds non-cached accesses, BUS_REQ_RD event can instead be approximated by BUS_ACCESS_RD/2.
- The BUS_ACCESS and BUS_ACCESS_WR events are not accurate.
- The count of *PERCYC events can appear to overflow at multiples of 256, making their counts much smaller than expected.

Workaround

Some of the affected events have workarounds, which are mentioned in the implications section.

For the remaining events, there are no workarounds.

2626173
ERR0STATUS.SERR might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The SERR field of the ERR0STATUS register might be updated incorrectly.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core executes an SVE non-fault, first-fault or predicated load instruction.
2. The core executes another load instruction, and this load follows the load above with a maximum of one other instruction in between.
3. Both of these two instructions see an external abort (NDerr or Derr) or poison data. The abort must be different between the two.
4. Timing-sensitive, micro-architectural conditions occur.

Implications

There is still substantial benefit being gained from the *Error Correction Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions are met, the RAS ERR0STATUS.SERR field might be set to 18 instead of the expected values of 12 or 21.

Workaround

No workaround is required.

2626511
Minimum power policy might prevent power off when FUNC_RET in use

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The core supports putting the VPU logic in a low power state with the FUNC_RET power mode. If the power mode is in use while the PPU minimum power policy mode is also set to the FULL_RET or FUNC_RET, then it can prevent the other core in the same complex from powering off.

Configurations Affected

This erratum affects all configurations with two cores in a complex.

Conditions

1. One core in the complex has the IMP_CPUPWRCTLR_EL1.VPU_PWR_CTLR register set to a nonzero value.
2. The core does not execute any instructions that use the VPU for a period and therefore enters the FUNC_RET power mode.
3. The IMP_CPUPWRCTLR_EL1.CORE_PWRDN_EN bit is set. This might happen before or after condition 2.
4. The core executes a WFI instruction.
5. The PPU for that core is in dynamic mode with the minimum power policy programmed to FULL_RET or FUNC_RET in the PPU_PWPR.PWR_POLICY field.
6. The other core in the complex requests a power transition from ON to OFF, ON to OFF_EMU, or OFF_EMU to OFF.

Implications

If the erratum occurs, the second core in the complex will not be able to complete its power transition until the first core has moved from FUNC_RET to ON. However, the first core will not be correctly indicating to the PPU that it needs to transition to ON, therefore, neither core will be able to proceed, which could lead to a system deadlock. If during this time the minimum power mode of the first core is changed to OFF or OFF_EMU then the first core will transition to that power mode, which will then allow the second to complete its transition.

Arm does not expect the minimum power policy to typically be set to FUNC_RET or FULL_RET if the core is going to want to power off, because that would prevent the core from powering off.

Workaround

For systems that never set the core's minimum power policy to FULL_RET or FUNC_RET, no workaround is necessary. If a workaround is required, then as part of the powerdown sequence, EL3 software should set the IMP_CPUPWRCTLR_EL1.VPU_PWR_CTLR field to zero before it sets the IMP_CPUPWRCTLR_EL1.CORE_PWRDN_EN field.

2628441
External aborts might result in a deadlock

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

A deadlock might occur as a result of a load accessing a cache line with an external abort.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. A load accesses a cache line that sees an external abort.
2. Timing sensitive conditions occur.

Implications

There is still substantial benefit being gained from the *Error Correcting Codes* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum. If the conditions are met, the load might deadlock.

Workaround

There is still substantial benefit being gained from the ECC logic. There might be a small increase in overall system failure rate due to this erratum.

To prevent this erratum from occurring during functional testing, software can set IMP_CPUACTLR_EL1[29] = 1, for example, using the following sequence:

```
MRS x0, S3_0_C15_C1_0
MOV x1, #1
BFI x0, x1, #29, #1
MSR S3_0_C15_C1_0, x0
```

This will reduce the effectiveness of internal clock gating, and might impact power efficiency. During power testing of sample silicon, Arm recommends not applying the workaround.

2637415
Core might not execute any instruction when performing Halting Step

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Halting Step is a debug resource that a debugger can use to make the core step through code one instruction at a time. Due to this erratum, the core might not execute any instruction before returning to debug state.

Configurations Affected

This erratum affects all configurations.

Conditions

This erratum occurs when the following conditions are true for the sequence described below:

- DBGEN == 1.
- SPIDEN == 0.
- Halting Step is enabled by EDECR.SS.

Then the following sequence must occur:

1. The core exits Debug State to Non-secure state.
2. The core must execute `ESB` instruction with a physical SError pending and unmasked according to the table in the "Asynchronous exception masking" section of the Arm Architecture Reference Manual Armv8, for A-profile architecture. The SError must generate an exception targeting EL3.
3. The core performs an ERET from EL3 to a Non-secure state.
4. The core starts executing an instruction that will not get an exception targeting EL3.

Implications

The instruction at step 4 of the above sequence should be executed, then the core should enter debug state. Instead, the core will enter debug state without executing that instruction.

The next time the core attempt to step that instruction, it will be executed, and then the core will enter debug state in the correct manner.

Workaround

This erratum has no workaround.

2640950
External 32-bit writes to some 64-bit RAS registers are not mapped correctly

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Writing to the upper or lower 32 bits of a 64-bit RAS register is permitted. Due to this erratum, some addresses are mapped to the incorrect range of bits inside a register.

In the core node, writes addressed to the upper 32 bits of the following registers update the lower 32 bits instead. The upper 32 bits are inaccessible by 32-bit writes.

- ERR0MISC0_EL1
- ERR0MISC1_EL1

The same issue applies to the following core node registers, but their upper 32 bits are RES0, so 32-bit writes to those addresses are not expected.

- ERR0CTLR_EL1
- ERR0PFGCDN_EL1
- ERR0PFGCTL_EL1
- ERR0STATUS_EL1

In the Complex node, writes addressed to the upper 32 bits of ERR0MISC1_EL1 update the lower 32 bits instead. Also, writes addressed to the lower 32 bits update the upper 32 bits instead.

Configurations Affected

All configurations are affected.

Conditions

This erratum occurs whenever there is a 32-bit external write to the upper 32 bits of the affected registers.

Implications

For the affected registers which contain error record information, it is not possible to clear them completely using only 32-bit external writes.
For the affected registers which control error records or pseudo-fault generation, the settings in the lower 32 bits can be corrupted.
All affected registers can still be read as expected using 32-bit reads from the assigned addresses.

Workaround

This erratum can be avoided by using 64-bit writes for the affected registers.

2668978
External aborts reporting cannot be disabled

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

A system able to generate External aborts or poison might see errors reported by the L2 node, even if error reporting is disabled.

Configurations Affected

This erratum affects configurations with CORE_CACHE_PROTECTION enabled (CORE_CACHE_PROTECTION set to 1).

Conditions

- ERR0CTRL.ED is set to 0b0
- Data or responses are received by the core with Data Error, Non-data Error, or poison

Implications

If the previous conditions are met, an error might be reported.
Interrupts generated by the set of conditions can be masked by clearing ERR0CTLR.FI and ERR0CTLR.UI.

Workaround

This erratum has no workaround.

2679529
Multiple simultaneous errors report for L1 data cache might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

If multiple *Error Correcting Code* (ECC) errors are simultaneously detected in the L1 data cache, the reported ERR0MISC0 and ERR0STATUS for the L1 data cache Data RAM or *Memory Tagging Extension* (MTE) data RAMs might be incorrect.

Configurations Affected

This erratum affects configurations where the CORE_CACHE_PROTECTION parameter is TRUE.

Conditions

This erratum occurs under the following conditions:

1. ECC errors are detected in multiple banks in the same cycle in either the L1 data cache data RAMs or the L1 data cache MTE data RAMs.
2. The ECC error happen in different index/way.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions are met, the value reported in the ERR0STATUS, ERR0MISC0.INDX and/or ERR0MISC0.WAY field in the core RAS register block might be incorrect. It can be either:

- The ERRSTATUS may not report the highest priority error of the simultaneous error for Data RAM, but the index/way information report is still correspond to the report error
- The ERRSTATUS report the highest priority error of the simultaneous error for MTE data RAM, but the index/way information may not corresponding to the highest priority error for the MTE data RAM.

Workaround

No workaround is required for this erratum.

2681778
TLBI not fully invalidating entries because of parity errors

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Entries might not be fully invalidated on TLBI because of parity errors.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

- The Complex receives a DVM TLBI.
- TLB prefetches are enabled.
- Parity errors occurring.
- Uncommon, timing-sensitive micro-architectural conditions occur.

Implications

If the previous conditions are met, a TLBI might not invalidate some entries in the TLB.
There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

Workaround

No workaround is expected to be required for engineering samples.

2690489
Some architectural PMU events are not always available to trace unit

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

The PMU architectural events are available to the trace unit through the extended input facility. Due to this erratum, some architectural events might not be sent to the trace unit after being selected. The affected events include most events above 0x4000:

- 0x4005 STALL_BACKEND_MEM
- 0x4006 L1I_CACHE_LMISS
- 0x4009 L2D_CACHE_LMISS_RD
- 0x400B L3D_CACHE_LMISS_RD
- 0x4020 LDST_ALIGN_LAT
- 0x4021 LD_ALIGN_LAT
- 0x4022 ST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_CHECKED_RD
- 0x4026 MEM_ACCESS_CHECKED_WR
- 0x8002 SVE_INST_RETIRED
- 0x8006 SVE_INST_SPEC
- 0x8014 FP_HP_SPEC
- 0x8018 FP_SP_SPEC
- 0x801C FP_DP_SPEC
- 0x80E3 ASE_SVE_INT8_SPEC
- 0x80E7 ASE_SVE_INT16_SPEC
- 0x80EB ASE_SVE_INT32_SPEC
- 0x80EF ASE_SVE_INT64_SPEC
- 0x810C BR_INDNR_TAKEN_RETIRED
- 0x8110 BR_IMMED_PRED_RETIRED
- 0x8111 BR_IMMED_MIS_PRED_RETIRED
- 0x8114 BR_RETURN_PRED_RETIRED
- 0x8115 BR_RETURN_MIS_PRED_RETIRED
- 0x8116 BR_INDNR_PRED_RETIRED
- 0x8117 BR_INDNR_MIS_PRED_RETIRED
- 0x811C BR_PRED_RETIRED
- 0x811D BR_IND_RETIRED
- 0x8120 INST_FETCH_PERCYC
- 0x8121 MEM_ACCESS_RD_PERCYC
- 0x8124 INST_FETCH
- 0x8125 BUS_REQ_RD_PERCYC

- 0x8128 DTLB_WALK_PERCYC
- 0x8129 ITLB_WALK_PERCYC
- 0x8134 DTLB_HWUPD
- 0x8135 ITLB_HWUPD
- 0x8136 DTLB_STEP
- 0x8137 ITLB_STEP
- 0x8138 DTLB_WALK_LARGE
- 0x8139 ITLB_WALK_LARGE
- 0x813A DTLB_WALK_SMALL
- 0x813B ITLB_WALK_SMALL
- 0x813C DTLB_WALK_RW
- 0x8145 L1I_CACHE_HWPRF
- 0x8154 L1D_CACHE_HWPRF
- 0x8155 L2D_CACHE_HWPRF
- 0x8156 L3D_CACHE_HWPRF
- 0x8158 STALL_FRONTEND_MEMBOUND
- 0x8159 STALL_FRONTEND_L1I
- 0x815B STALL_FRONTEND_MEM
- 0x815C STALL_FRONTEND_TLB
- 0x8160 STALL_FRONTEND_CPUBOUND
- 0x8161 STALL_FRONTEND_FLOW
- 0x8162 STALL_FRONTEND_FLUSH
- 0x8164 STALL_BACKEND_MEMBOUND
- 0x8165 STALL_BACKEND_L1D
- 0x8167 STALL_BACKEND_TLB
- 0x8168 STALL_BACKEND_ST
- 0x816B STALL_BACKEND_BUSY
- 0x816C STALL_BACKEND_ILOCK
- 0x818D BUS_REQ_RD
- 0x81BC L1D_CACHE_REFILL_HWPRF
- 0x81BD L2D_CACHE_REFILL_HWPRF
- 0x82FA DTLB_WALK_HWPRF

Configurations Affected

All configurations are affected.

Conditions

- Trace unit is configured to use the extended input facility with an affected event
- The affected event is not enabled for counting in the PMU through the PMEVTYPER<n>EL0 registers

Implications

The affected events cannot be used reliably by the trace unit unless the PMU is also configured to count the same event.

Workaround

This erratum can be avoided if the PMU is configured to count the event selected by the trace unit.

2708967
Read value of PMMIR is incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0. Fixed in r0p1.

Description

Due to this errata, the BUS_SLOTS and BUS_WIDTH fields of the PMMIR_EL1 system register read as 0 instead of their documented intended values, 0b0010 and 0b0110 respectively. All other fields have their intended value. The incorrect values are also read from the memory-mapped external register, PMMIR.

Configurations Affected

All configurations are affected.

Conditions

The values read for these fields are always incorrect.

Implications

A 0 value in these fields is valid and indicates that the information is not available to software. Software has less information about the BUS_ACCESS performance event than intended.

Workaround

This erratum has no workaround.

2710075
Read value of IMP_CPUCFR_EL1 might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Due to this errata, several fields of the IMP_CPUCFR_EL1 system register always read as 0 instead of their intended value. The affected fields are:

- L2_RAM_EVA
- L2_NUM_RAMCTL_PARTITIONS
- L2_NUM_TAGCTL_SLICES

Configurations Affected

The values in these fields will match configurations where L2_RAM_EVA is FALSE, and L2_NUM_RAMCTL_PARTITIONS and L2_NUM_TAGCTL_SLICES are both set to 1. All other configurations will have an incorrect value at least one of these fields.

Conditions

For affected configurations, IMP_CPUCFR_EL1 fields with an incorrect value will always be incorrect.

Implications

The information provided by this register might not match the configuration of the L2 cache. Software is not expected to rely on these values.

Workaround

There is no workaround.

2713358
ERRxSTATUS.UET field might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

ERR2STATUS register contain information about reported errors. Under some conditions, the value of the UET field might be incorrect.

Configurations Affected

All configurations are affected.

Conditions

If CORE_CACHE_PROTECTION = 'b1:

- An external abort is detected on a clean, L2 cache allocating line.
- At the same time, a double-bit error is detected on L2 TAGRAMs or L2 L1 Duplicate TAGRAMs.

If CORE_CACHE_PROTECTION = 'b0:

- An external abort is detected on a clean, L2 cache allocating line.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur with CORE_CACHE_PROTECTION = 'b1, the reported error might record UEO instead of UC on ERR2STATUS.UET.
If the condition occur with CORE_CACHE_PROTECTION = 'b0, the reported error might record UC instead of UEO on ERR2STATUS.UET.

Workaround

There is no workaround required.

2713644
Cache debug target for L2 Data RAM may not record correct data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Cache debug target for L2 data RAM might not record the correct data for the top half of a cache line.

Configurations Affected

All configurations with CORE_CACHE_PROTECTION set to False are affected.

Conditions

Software executes a cache debug read targeting the L2 data RAM using SYS #6, C15, C4, #3{, <Xt>}.

Implications

The cache debug data recorded in the IMP_CDBGDR0_EL3 register for the L2 data RAM top half cache line will always be 0, and not reflect the value in the data RAM.

Workaround

There is no workaround required.

2732181
ERR2STATUS might be incorrect

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The ERR2STATUS register contains information about reported errors. Under some conditions, the value of the register might be incorrect.

Configurations Affected

All configurations are affected.

Conditions

If CORE_CACHE_PROTECTION = FALSE:

- A L2 allocating line is received with poison.

If CORE_CACHE_PROTECTION = TRUE:

- A write to the ERR2STATUS register is performed.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the condition with CORE_CACHE_PROTECTION = FALSE is met:

- The PN field is incorrectly set to 0.

If the condition with CORE_CACHE_PROTECTION = TRUE is met:

- The write is performed while the OF field is set and the write is not clearing it.

Workaround

There is no workaround required.

2740664
PMU event 0x77 CRYPTO_SPEC does not always count when enabled

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The PMU event 0x77 CRYPTO_SPEC does not always count when enabled.

Configurations Affected

This erratum affects all configurations.

Conditions

Under the following condition, PMU event 0x77 CRYPTO_SPEC will never count:

- PMU is configured to count event 0x77, and no events in the range 0x78 to 0xBF are enabled for counting.

Otherwise, the event is counted as expected.

Implications

PMU event 0x77 CRYPTO_SPEC cannot be used without using the software workaround.

Workaround

This erratum can be avoided if the PMU is configured to count an event in the range 0x78 to 0xBF.

2751027
Load operation might abort unexpectedly when accessing poisoned data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Load (or atomic load) might abort data accesses when the operation does not access the poisoned data, but the 256 bits aligned data it accesses contain poisoned data.

Configurations Affected

This erratum affects all configurations.

Conditions

1. The core executes a load (or atomic load) operation
2. The load operation miss in L1 cache
3. A part of the returned data contains poisoned data

Implications

If the previous conditions are met, the load (or atomic load) instruction might generate synchronous external Data Abort, even if the chunk of data it accesses does not contain poisoned data.
For atomic load instruction, this means the memory might be updated by the atomic instruction but it still generates precise abort for the atomic instruction.

Workaround

No workaround is required for this erratum.

2803663
Speculative dirty bit hardware update might happen for store operation

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Speculative dirty bit hardware update might happen during a store operation.

Configurations Affected

This erratum affects configurations where the CORE_CACHE_PROTECTION parameter is TRUE.

Conditions

This erratum occurs under the following conditions:

1. Load operation A
2. Store operation B, following load operation A
3. Hardware update of the dirty bit is enabled for the page of memory accessed by the store operation B
4. Load operation A encounters a potentially correctable ECC error in the L1 data cache, the load operation is microarchitecturally replayed. The erratum occurs If during the replay, the load sees an abort that was not present in the original execution.
5. Timing sensitive microarchitectural condition happens

Implications

If the previous conditions are met, the Store operation might update the AP[2] or S2AP[1] bit as writable when it should not.

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

Workaround

No workaround is required for this erratum.

2833401
Direct access to internal memory might not be reliable

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Internal memory used by structures in the L2 cache can be read using system registers. In some conditions, the read value might not reflect the value stored in the memory.

Configurations Affected

This erratum affects configurations having the parameter CORE_CACHE_PROTECTION set to TRUE.

Conditions

1. A core is active.
2. A stream of cache debug operations is issued, using system registers IMP_CDBGL2CMR and IMP_CDBGL2CDR, while the memory system is processing loads and stores to Normal memory.
3. An *Error Correcting Code* (ECC) error is detected in the L2 DATA RAM.
4. Complex microarchitectural timing conditions occur.

Implications

Arm expects that the memory system is in a quiescent state while direct access to memory is being performed.
If the conditions occur, the values read while directly accessing internal memory might not be correct.

Workaround

No workaround is required.

2841875
An uncontainable error might deadlock the cluster

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

An uncontainable error detected when a core is doing a line upgrade might deadlock the cluster.

Configurations Affected

This erratum affects all configurations.

Conditions

1. A core is doing a store that requires a line upgrade.
2. Unlikely, timing-sensitive, microarchitectural conditions occur, including an uncontainable error detected in the L1 Duplicate Tag RAMs.

Implications

There is still substantial benefit being gained from the *Error Correcting Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur, the complex might deadlock.

Workaround

There is no workaround.

2853709
Error record registers indicate incorrect feature support in configurations without cache protection

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The following error record feature registers contain information about the features implemented in the corresponding RAS node. The affected nodes are Node 1 (L1 memory system) and Node 2 (L2 memory system).

- **ERR1FR**
- **ERR1PFGF**
- **ERR2FR**
- **ERR2PFGF**

In configurations without cache protection, these registers incorrectly indicate that cache protection is present. This also applies when reading the **ERXFR_EL1** and **ERXPFGF_EL1** registers while **ERRSELR_EL1** is selecting any of the affected nodes. Indication of support for error types caused by External aborts is unaffected.

Also, in the **ERR1PFGCTL** and **ERR2PFGCTL** pseudo-fault generation control registers, fields which control the injection of cache protection error types can be read and written as if support is present. Attempting to inject an unsupported error type will have no effect on pseudo-fault generation.

Configurations Affected

All configurations with the **CORE_CACHE_PROTECTION** parameter set to FALSE are affected.

Conditions

The incorrect read and write behavior of these registers in affected configurations is always present.

Implications

In configurations without cache protection, software might incorrectly assume from these register values that it is actually present. This might lead to an unexpected error injection test result, if a test attempts to inject an error type which it thinks is supported, but no error record is created.

Workaround

This erratum has no workaround. Contact Arm for more details on which error types are not supported when cache protection is not present.

2861633
Some PMU events are incorrectly masked to trace unit

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p1. Fixed in r0p2.

Description

Common architectural and microarchitectural *Performance Monitoring Unit* (PMU) events are available to the trace unit through the extended input facility. Due to this erratum, under certain conditions, some events might not be sent to the trace unit after being selected:

- 0x4020 LDST_ALIGN_LAT
- 0x4021 LD_ALIGN_LAT
- 0x4022 ST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_CHECKED_RD
- 0x4026 MEM_ACCESS_CHECKED_WR

Configurations Affected

All configurations are affected.

Conditions

The erratum occurs when all of the following conditions are met:

- Trace unit is configured to use the extended input facility with an affected event.
- Performance Monitors are disabled because they have been configured to be disabled at EL3 or in Secure state.
- Self-hosted trace is enabled.

Implications

The affected events cannot be used reliably by the trace unit at EL3 or in Secure state unless the Performance Monitors are enabled.

Workaround

There is no workaround.

2871911
LDG or MTE checked load/store might fail to detect poisoned data

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

An LDG or MTE checked load/store might fail to detect poisoned data.

Configurations Affected

This erratum affects configurations with BROADCASTMTE = TRUE.

Conditions

The erratum occurs under the following conditions:

1. An older STG or store operation to cache line X.
2. A younger LDG or MTE checked load/store to the same cache line X.
3. Cache line X has a deferred error.
4. Timing sensitive, microarchitectural conditions occur.

Implications

If the conditions are met, the LDG or MTE checked load/store might fail to detect poisoned data, and might return an incorrect result for an LDG, or an incorrect tag check result for a checked load or store. If asynchronous MTE tag checks are enabled, the state of TFSR_ELx might get corrupted.

Workaround

No workaround is required for this erratum.

2872870
CE or DE errors from L1 data cache access might not be recorded in the RAS records

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

A corrected or deferred error detected during an access to the data or MTE RAMs of the L1 data cache might not be reported to RAS.

Configurations Affected

This erratum affects all configurations with parameter CORE_CACHE_PROTECTION set to TRUE.

Conditions

1. A line is cached in L1 in shared clean state
2. An error is detected in the data or MTE RAMs of the L1 data cache while accessing the line.

Implications

There is still substantial benefit being gained from the *Error Correcting Code* (ECC) logic. There might be a negligible increase in overall system failure rate due to this erratum.
If the conditions occur, CE and DE detected when accessing the data or MTE RAMs of the L1 data cache will not be recorded in the RAS node. Error correction functionality and aborts handling are not affected by this erratum.

Workaround

There is no workaround.

2879977
Unmodified cache line might be written back to memory

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

A cache line might be temporarily marked as modified, which might result in that line being written back to memory.

Configurations affected

This erratum affects all configurations with parameter BROADCASTMTE set to True.

Conditions

This erratum occurs under the following conditions:

1. Memory location A is marked as Normal Inner Write-Back, Outer Write-Back Cacheable memory.
2. The core allocates location A into the L1 data cache or the L2 cache in Unique state without MTE tags. This allocation might be due to committed instructions, speculative execution, or data prefetching.
3. The core executes an STG, ST2G, or STGM that requires fetching of MTE tags by the L2.
4. Another *Request Node* (RN) is requesting the line, and this request is ordered before the previous condition (3) in the *Home Node* (HN). The core will provide the line as dirty, but data remains unchanged.

Implications

If the previous conditions are met, the cache line for memory location A might be marked as modified, but the data remains unchanged.
If the line is evicted to memory while set as dirty with unchanged data, it will then overwrite the value in memory. If an agent in the system has written to location A through a Non-cacheable mapping, these writes might then be overwritten with the older data from the core cache write-back, causing these writes to no longer be visible. This might then result in data corruption for software-managed coherency use cases.
The scenario is a race conditions where an old value of location A can be temporarily seen by a Non-cacheable observer. If the core executes a read to memory location A before the store, that is requiring a DC IVAC (Data or unified Cache line Invalidate by VA to PoC), the race condition is resolved, and the erratum is not applicable.

Workaround

No workaround is required for this erratum.

2940628
Store data might be lost when a correctable error is detected in the L1 data cache

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Store data might be lost when a correctable error is detected in the L1 data cache RAM or L1 data cache *Memory Tagging Extension* (MTE) tag RAM.

Configurations affected

This erratum affects configurations with CPU_CACHE_PROTECTION set.

Conditions

This erratum occurs under the following conditions:

1. The core executes two or more stores to the same cache line but to different 16B-aligned quantities. At least one of the stores is a store of less than 32 bits, or is MTE tag-checked.
2. The partial or MTE tag-checked store above sees a correctable *Error Correcting Code* (ECC) error in the L1 data cache RAM or L1 data cache MTE tag RAM.
3. Very unlikely, timing-sensitive micro-architectural conditions occur.

Implications

There is still substantial benefit being gained from the ECC logic. There might be a negligible increase in overall system failure rate due to this erratum.

If the conditions are met, a single-bit error could result in silent data corruption. This is similar to the case of a triple-bit error incorrectly being detected as a single-bit error.

One of the stores at condition (1) can write to the cache without marking the cache line as dirty. A second store that must already be in the store buffer will shortly mark the line dirty, but the line can be lost from the *Processing Element* (PE) caches in the small window between both operations. If the cache line leaves the PE caches in this window, the data from the store that has already been written to the cache might be lost, as it is not marked as dirty.

Workaround

No workaround is required.

2969138
Unmodified page table cache lines might be written back to memory

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

Hardware management of the Access Flag or Dirty State might set a cache line containing page table data as dirty even if the descriptor data has not been modified.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. Hardware management of the Access Flag is enabled (TCR_ELx.HA or VTCR_EL2.HA are set to 0b1) or Hardware management of the Dirty State is enabled (TCR_ELx.HD or VTCR_EL2.HD are set to 0b1).
2. The *Processing Element* (PE) schedules an Access Flag or Dirty State update after having read a descriptor.
3. The descriptor changes before the hardware update is performed, and one of the following conditions applies to the new descriptor:
    - It is not a page or block descriptor anymore.
    - It does not have the DBM bit set.
    - It has different permissions than the old descriptor.

Implications

If the conditions are met, the descriptor will not be modified, but the line will be marked as dirty. If the stage 1 hardware update of the access flag or dirty bit fails because of a stage 2 fault, the issue does not occur.

The cache line will be written back to main memory when evicted from the PE, which can result in a negligible increase in system power.

Workaround

No workaround is required.

3006395
PMU Events might be inaccurate

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

The following performance events might be unreliable due to this erratum:

- 0x0015 L1D_CACHE_WB
- 0x0018 L2D_CACHE_WB
- 0x0020 L2D_CACHE_ALLOCATE
- 0x0024 STALL_BACKEND
- 0x002B L3D_CACHE
- 0x0039 L1D_CACHE_LMISS_RD
- 0x003C STALL
- 0x0043 L1D_CACHE_REFILL_WR
- 0x0053 L2D_CACHE_REFILL_WR
- 0x0077 CRYPTO_SPEC
- 0x00A2 L3D_CACHE_REFILL_RD
- 0x00E5 STALL_BACKEND_ILOCK_ADDR
- 0x4005 STALL_BACKEND_MEM
- 0x400E TRB_TRIG
- 0x8014 FP_HP_SPEC
- 0x8018 FP_SP_SPEC
- 0x801C FP_DP_SPEC
- 0x80E3 ASE_SVE_INT8_SPEC
- 0x80E7 ASE_SVE_INT16_SPEC
- 0x80EB ASE_SVE_INT32_SPEC
- 0x80EF ASE_SVE_INT64_SPEC
- 0x8158 STALL_FRONTEND_MEMBOUND
- 0x8159 STALL_FRONTEND_L1I
- 0x815B STALL_FRONTEND_MEM
- 0x8165 STALL_BACKEND_L1D

Configurations affected

For the STALL_FRONTEND_* events and L2D_CACHE_* events, this erratum affects all configurations with L2 cache. For the other events, this erratum affects all configurations.

Conditions

No specific conditions are needed for this erratum.

Implications

The affected events have been divided into categories:

- Events in the Low impact category still produce an indicative result in most cases.
- Events in the High impact category are likely to produce misleading inaccurate results.

Low impact:

- The L3D_CACHE event might slightly overcount.
- The L1D_CACHE_WB event might slightly overcount.
- The L2D_CACHE_WB event might overcount, by incorrectly including transfers of clean data.
- The L1D_CACHE_REFILL_WR event might slightly undercount.
- The STALL event might slightly undercount
- The CRYPTO_SPEC event might overcount, by incorrectly including speculative execution of 64-bit-to-128-bit PMULL variants
- The FP_HP_SPEC event might overcount, by incorrectly including speculative execution of a small number of integer instructions or FP conversions from wider precisions
- The FP_SP_SPEC and FP_DP_SPEC events might undercount, by incorrectly excluding speculative execution of some FP conversions from single or double precision to half precision, or overcount, by incorrectly including speculative execution of some integer instructions
- The ASE_SVE_INT8_SPEC event might overcount, by incorrectly including speculative execution of 64-bit-to-128-bit PMULL variants and some 8-bit-to-16-bit PMUL variants
- The ASE_SVE_INT16_SPEC event might undercount, by incorrectly excluding speculative execution of some 8-bit-to-16-bit PMUL variants
- The ASE_SVE_INT32_SPEC and ASE_SVE_INT64_SPEC events might overcount, by incorrectly including a small number of single-precision and double-precision FP instructions

High impact:

- The STALL_FRONTEND_L1I and STALL_FRONTEND_MEM events might significantly undercount.
- The L2D_CACHE_ALLOCATE event might significantly overcount.
- The L1D_CACHE_LMISS_RD event might be significantly undercount.
- The L2D_CACHE_REFILL_WR event might overcount, by incorrectly including non-L2 allocating stores.
- The L3D_CACHE_REFILL_RD event might significantly undercount.
- The STALL_BACKEND_L1D event might significantly undercount, causing STALL_BACKEND_MEM to overcount.
- The STALL_BACKEND event might significantly undercount or overcount.
- The STALL_BACKEND_ILOCK_ADDR event might significantly undercount.
- The TRB_TRIG event might significantly overcount.

Workaround

STALL_BACKEND will count most consistently when STALL_SLOT_BACKEND, and one or more other types of stall backend event (such as STALL_BACKEND_MEM) are also enabled.
Otherwise, there are no workarounds. The overall impact of inaccuracies in these PMU events is considered to be low enough that workarounds should not be necessary.

3067972
AMU does not count STALL_BACKEND_MEM correctly

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

AMU counter AMEVCNTR03 should count PMU event 0x4005 STALL_BACKEND_MEM. Due to this errata, AMEVCNTR03 counts more often than the PMU event. The count results are approximately equivalent to the PMU event 0x8164 STALL_BACKEND_MEMBOUND.

Configurations affected

All configurations are affected

Conditions

No specific conditions are needed for this erratum.

Implications

This inconsistency is not expected to have any impact in the intended AMU use case for DVFS. The count will be larger, but is still based on stalls caused by memory activity.

Workaround

No workaround should be necessary, because avoiding this event should have very low impact. If software does use this event, the larger count should not significantly hinder DVFS implementations. In cases where the PMU is enabled, then 0x4005 STALL_BACKEND_MEM, or a similar alternative, can be counted in the PMU instead.

3094433
The core might lose register accesses or interrupts in low-power state

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

Description

In rare circumstances, a core in low-power state might lose debug accesses, system register accesses or interrupts. In addition, microarchitectural accesses related to reset might be lost. This might lead to a system deadlock.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. At least one core in a complex is in the low-power state or Off power mode.
2. At least one other component in the system is active and driving system register or interrupt traffic through the complex.
3. One of the following events occurs, targeting the core in low-power state:
    - Warm or power-on reset
    - External Debug register access
    - System register access through the Utility bus
    - An interrupt
4. Uncommon, timing-sensitive microarchitectural conditions occur.

Implications

If the conditions are met, then the core in low-power state might have its clock automatically gated during any of the mentioned events, which might lead to a system deadlock.
The timing-sensitive conditions are expected to be rare. The system will recover in most cases when a subsequent event causes the clock to be restored, allowing the first event to complete, resolving the deadlock.

Workaround

Due to the rarity of the conditions, and the potential for a subsequent event to break the deadlock, Arm does not believe that a workaround is required.

3103429
STREX might raise both Synchronous and Asynchronous abort for external abort

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0 and r0p1. Fixed in r0p2.

Description

STREX might raise both Synchronous and Asynchronous abort for external abort.

Configurations affected

This erratum affects all configurations.

Conditions

1. The core executes a store exclusive to shareable memory other than Inner-Writeback and Outer-Writeback.
2. The store exclusive operation sees a DErr or NDErr response from a downstream interconnect.

Implications

If the previous conditions are met, the store exclusive takes the external abort as a synchronous data abort exception, but will also signal an SError interrupt.

Workaround

No workaround is required for this erratum.

3145557
Some PMU events for Operation speculatively executed do not count correctly

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

PMU event 0x0073 DP_SPEC incorrectly does not include the following instruction types:

- System register instructions
- System instructions
- Hint instructions
- PAC instructions

PMU event 0x0071 ST_SPEC incorrectly includes some memory related barriers, system registers, and system instructions.

PMU event 0x0076 PC_WRITE_SPEC has the following inaccuracies:

- Includes ISB instructions while 0x000C PC_WRITE_RETIRED does not. The inclusion of ISB in both events is implementation defined. These two events should be consistent, and the intention was that ISB would not be included.
- System register writes which include a Context Synchronization Event (such as direct writes to PSTATE) incorrectly count this event.
- ERET instructions incorrectly do not count this event.

Configurations affected

This erratum affects all configurations.

Conditions

No specific conditions are needed for this erratum.

Implications

Extra consideration should be taken when comparing PMU results for these events with other CPUs. The overall impact should be low because barriers, system register instructions, system instructions, and hints are expected to be a low percentage of code. The effect of PAC instructions being missing from DP_SPEC could have slightly higher impact, depending on the routine being measured.

Workaround

There is no workaround for this erratum. A workaround should not be necessary.

3185964
A watchpoint event might not be generated for a tag-checked SVE non-fault or first-fault load instruction

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

Description

A watchpoint event might not be generated correctly for a tag-checked SVE non-fault or first-fault load instruction.

Configurations affected

This erratum affects configurations with BROADCASTMTE = true.

Conditions

This erratum occurs under the following conditions:

1. MTE checking is enabled (SCTLR_ELx.ATAn = 1, SCTLR_ELx.TCFn != 0b00).
2. The core executes a checked SVE non-fault or first-fault load instruction.
3. The load instruction crosses a 16-byte boundary, but does not cross a page boundary.
4. The load above fails the MTE tag check on the 16-byte-aligned quantity that crossed over the 16-byte boundary.
5. Watchpoints are enabled, and set to a memory location covered by the load above. For a first-fault load, the watchpoint is not covering the first accessed element.
6. Uncommon micro-architectural conditions occur.

Implications

If the conditions are met, the watchpoint event might not be generated, and the FFR register might not reflect that a watchpoint event occurred.

Workaround

No workaround is required for this erratum.

3186696
The virtual address is not sign extended in EDWAR

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

Description

EDWAR returns the virtual data address being accessed when a Watchpoint Debug Event was triggered. Virtual addresses are 64-bit in AArch64.

Virtual addresses reported in EDWAR[63:49] get truncated and are padded with zeros instead of sign extended.

The valid VA range for each TTBR is maximum of 48-bits (this core does not support FEAT_LVA), which is why VA was getting microarchitecturally truncated to save some flops.

In EL1+EL0 and EL2+EL0 translation regimes, there are two TTBRs. Addresses in the TTBR1 range are 0xFFFFFFFFFFFFFFFF down to the size of the TTBR range (so down to 0xFFFF000000000000 if using the full 48-bit range).

With the truncation+ZeroPadding, if there is a watchpoint on the address 0xFFFF000000000000, then EDWAR will report 0x000100000000000, which violates the architecture.

Configurations affected

This erratum affects all configurations.

Conditions

1. Watchpoint is enabled by DBGWCR<n>_EL1 register.
2. Virtual Address in DBGWVR<n>_EL1 belongs to TTBR1 range.

Implications

Virtual address reported in EDWAR[63:49] will be incorrect.

Workaround

After reading EDWAR, Debugger can perform something as below:

```
if(EDWAR[48]==1) {
```

```
    EDWAR = EDWAR | 0xFFFE000000000000;
}
```


3311443
External registers PMCEID2 and PMCEID3 are not implemented

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

Description

External registers PMCEID2 and PMCEID3 should be accessible using the external interface to the Performance Monitors registers. These registers should be mapped as follows to System registers:

- PMCEID2 to PMCEID0_EL0[63:32]
- PMCEID3 to PMCEID1_EL0[63:32]

Due to this erratum, reads of these external registers always return 0.

Configurations affected

This erratum affects all configurations.

Conditions

An external read of PMCEID2 or PMCEID3 occurs.

Implications

These registers are read-only and indicate whether certain *Performance Monitoring Unit* (PMU) events are implemented, using one bit per event. Reading a 1 indicates the event is implemented. The following events are implemented, but their corresponding bit will read as 0:

- 0x4005 STALL_BACKEND_MEM
- 0x4006 L1I_CACHE_LMISS
- 0x400C TRB_WRAP
- 0x400E TRB_TRIG
- 0x4010 TRCEXTOUT0
- 0x4011 TRCEXTOUT1
- 0x4012 TRCEXTOUT2
- 0x4013 TRCEXTOUT3
- 0x4018 CTI_TRIGOUT4
- 0x4019 CTI_TRIGOUT5
- 0x401A CTI_TRIGOUT6
- 0x401B CTI_TRIGOUT7

- 0x4020 LDST_ALIGN_LAT
- 0x4021 LD_ALIGN_LAT
- 0x4022 ST_ALIGN_LAT
- 0x4024 MEM_ACCESS_CHECKED
- 0x4025 MEM_ACCESS_RD_CHECKED
- 0x4026 MEM_ACCESS_WR_CHECKED

The following events are only implemented in certain configurations. In configurations where they are present, they are affected by this erratum:

- 0x4009 L2D_CACHE_LMISS_RD
- 0x400B L3D_CACHE_LMISS_RD

Workaround

The correct information can be accessed through the System registers PMCEID0_EL0 and PMCEID1_EL0, and is documented in the *Technical Reference Manual* (TRM).

3326745
Power transition from EMU_OFF to OFF might not complete

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

If debug operations are made to a core that is transitioning from EMU_OFF to OFF power mode, then the transition might not complete.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. The core is in the EMU_OFF power mode
2. PPU_PWPR.PWR_POLICY is re-programmed to bring this core to OFF power mode (from 0b0001 to 0b0000)
3. The external debugger makes an access to the core
4. Additional microarchitectural timing conditions occur

Implications

If the previous conditions are met, the power transition to OFF will not complete.

Workaround

This erratum is not expected to be seen in production devices as the expected use case is during development stages. If required, debugger can re-program PPU_PWPR.PWR_POLICY to OFF state after debug operations are completed.

3445783
Some PMU events do not count correctly

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1 and r0p2. Fixed in r0p3.

Description

Some PMU events have notable inaccuracies. Where applicable, any consistent architectural inaccuracies have been described. Otherwise, the inaccuracies are inconsistent or occur due to microarchitectural conditions.

- 0x0006 LD_RETIRED - counts atomic instructions which do not return data to the PE, such as STADD. Missing atomic instructions which return data, such as LDADD and CAS.
- 0x0007 ST_RETIRED - missing atomic instructions which do not return data and DC ZVA instructions
- 0x000B CID_WRITE_RETIRED - counts reads of CONTEXTIDR_EL1
- 0x006F STREX_SPEC
- 0x0070 LD_SPEC - counts some system register reads
- 0x0071 ST_SPEC - counts some system register writes, barriers, and cache maintenance operations

Configurations affected

This erratum affects all configurations.

Conditions

The conditions will depend on the event being counted.

Implications

Some PMU events will give inaccurate results.

Workaround

There are no workarounds for these events. ARM believes it is unlikely that a workaround would be necessary. If the precise accuracy of the mentioned events is a concern, please contact ARM.

3531773
TRBE might record the wrong status/syndrome information in the TRBSR_EL1 register

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

The Trace Buffer Unit might record the wrong status/syndrome information in TRBSR_EL1 register when a translation abort happens.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. The Trace Buffer is enabled
2. TRBLIMITR_EL1.FM !=2'b11
3. The trace buffer writes to a VA that encounters a translation fault

Implications

If the previous conditions are met two situations can occur, with TRBSR_EL1.wrap bit set even if no write data wrap event happens in both cases:

- The status/syndrome information recorded in the TRBSR_EL1 register might not correspond to the translation fault.
- The TRBSR_EL1 register record as translation fault.

The TRBPTR_EL1 Trace Buffer Write Pointer Register still records the correct page where the translation fault happens.

Workaround

ARM does not expect the condition 3 to happen, no workaround is required for this erratum.

3542363
Load/Store instruction might get unexpected translation fault

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A load/store instruction might get an unexpected translation abort when executed together with a PRFM instruction or a mis-predicted non-fault load instruction, when the TCR_ELx.NFDy bit is set.

Configurations affected

This erratum affects all configuration.

Conditions

This erratum occurs if one of the 2 following sequences happen.
First possible sequence of conditions:

1. Core running in EL0
2. The TCR_ELx.NFDy bit is set
3. PRFM instruction followed by Load/Store/CMO instruction which shares the same translation as the PRFM instruction
4. Uncommon micro-architecture condition happens

Second possible sequence of conditions:

1. Core running in EL0
2. The TCR_ELx.NFDy bit is set
3. Unaligned Load/Store instruction cross page boundary, followed by conditional or unconditional branch instruction
4. followed by a mis-predicted SVE non-fault contiguous load instruction or PRFM instruction which share the same translation as the above mentioned Load/Store
5. Uncommon micro-architecture condition happens

Implications

If the previous conditions are met, the Load/Store/CMO instruction might generate unexpected L0 level translation fault.

Workaround

There is no workaround required for this erratum.

3600964
A Non-cacheable store exclusive instruction receiving an NDErr or DErr response might update memory and raise synchronous abort

Status

Fault Type: Programmer Category C
Fault Status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A Non-cacheable store exclusive instruction receiving an NDErr or DErr response will raise a synchronous abort, but it might also update the memory.

Configurations affected

This erratum affects all configurations.

Conditions

This erratum occurs under the following conditions:

1. Store exclusive instruction to shareable memory other than Inner-Writeback and Outer-Writeback.
2. The write comp response for this store exclusive instruction returned with an NDerr or Derr error response.

Implications

If the previous conditions are met, the store exclusive instruction will raise synchronous abort, but the memory might also be updated by the store exclusive instruction.

Workaround

There is no workaround required.

3604547
Some PMU events do not count correctly

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

Some PMU events have notable inaccuracies. Where applicable, any consistent architectural inaccuracies have been described. Otherwise, the inaccuracies are inconsistent or occur due to microarchitectural conditions.

- 0x0001 L1I_CACHE_REFILL - counts for cache misses while instruction cache is disabled
- 0x0070 LD_RETIRED - counts atomic instructions which do not return data to the PE, such as STADD. Missing atomic instructions which return data, such as LDADD and CAS
- 0x0071 ST_RETIRED - missing atomic instructions which do not return data and DC ZVA instructions
- 0x00EE IMP_STALL_SLOT_BACKEND_ILOCK - counts stalls caused by interlock after dispatch instead of interlock cycles per slot at dispatch
- 0x8164 STALL_BACKEND_MEMBOUND - counts when there are no micro-operations available to dispatch

Configurations affected

This erratum affects all configurations.

Conditions

The conditions will depend on the event being counted.

Implications

Some PMU events will give inaccurate results.

Workaround

There are no workarounds for these events. ARM believes it is unlikely that a workaround would be necessary. If the precise accuracy of the mentioned events is a concern, please contact ARM.

3650470
Core in a dual core complex might complete power transition to OFF while a CTI trigger is in progress

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

A core in dual core complex might complete a power transition to OFF while a CTI trigger is in progress, which might cause a deadlock.

Configurations affected

This erratum affects all configurations with two cores in a complex.

Conditions

The erratum occurs under the following conditions:

1. The CTIs are enabled to generate triggers
2. A core makes one of the following power mode transitions:
    - ON to OFF
    - OFF_EMU to OFF
3. The CTI generates a trigger that targets at least both cores in the complex

Implications

If the previous conditions occur, the core might power transition to OFF while a CTI trigger is in progress. This will prevent transactions from completing on the debug APB interface, which might limit the debug capability on the system or cause a system deadlock.

Workaround

When cross triggers are in use for debugging, then the core can program DBGPRCR_EL1.CORENPDRQ to 1'b1 so that the core enters into the OFF_EMU power mode rather than the OFF mode.

3660280
CPU might incorrectly wake up from WFE state due to a virtual event stream when in FULL_RET power mode

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

Core incorrectly wakes up from WFE state due to a virtual event stream.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. The pseudocode function EL2Enabled() is TRUE, i.e. SCR_EL3.NS == '1' or SCR.EL3.EEL2 == '1'
2. HCR_EL2.<E2H,TGE> == '11'
3. CNTKCTL_EL1.EVNTEN == '1'
4. The core executes a WFE or WFET instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then the virtual event stream using the values of the CNTKCTL_EL1.{EVNTEN, EVNTDIR, EVNTI, EVNTIS} fields is enabled to wakeup the core from WFE or WFET state.

Workaround

No workaround is expected to be required for this erratum. If one is needed, disable entering full retention mode by setting both IMP_CPUPWRCTLR_EL1.WFE_RET_CTRL and IMP_CPUPWRCTLR_EL1.WFI_RET_CTRL to '000'.

3680961
Core might not execute some instructions in debug state after a reset catch debug event is generated

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

Under certain microarchitectural conditions, a *Processing Element* (PE) might not execute some instructions pushed via an external debugger to the EDITR. An external debugger might also observe a corrupted EDSCR.STATUS value.

Configurations Affected

This erratum affects all configurations.

Conditions

The erratum occurs after the following sequence:

1. Halting debug is not allowed.
2. The core exits reset state and a reset catch debug event is generated.
3. Halting debug is allowed.
4. The core enters debug state for another reason.
5. Certain microarchitectural conditions apply.

Implications

Under the above conditions, instructions pushed by an external debugger to the core via the EDITR might not be executed. The EDSCR.STATUS value might also appear corrupted.

Workaround

No workaround is expected to be required for this erratum.

3685405
The wrong shareability might be selected for instruction fetches when HCR_EL2.FWB == 1 while accessing a Non-shareable memory region

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

In certain architectural states that include HCR_EL2.FWB == 1, and accessing Non-shareable memory region, the core might use the wrong shareability for instruction fetch accesses.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following conditions are met:

- Core is running in EL1 or EL0 and using EL1&0 translation regime
- Stage 1 and Stage 2 MMU are on
- EL2 is enabled
- HCR_EL2.{FWB,ID,CD} = {1,0,0}
- SCTLR_EL1.{I,C} = {0,0}
- The combined S1 and S2 shareability is Non-shareable

Implications

If the conditions of the erratum are met, instruction accesses will use Non-shareable attributes while the data accesses will use Outer Shareable attributes. However, Arm does not expect hypervisors to configure memory as Non-shareable at Stage 2. Hence the implications of the erratum are not expected to be problematic.

Workaround

No workaround is expected to be required for this erratum.

3701090
IRG produces biased tag generation when GCR_EL1.RRND == 1

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

The algorithm used by the IRG instruction when GCR_EL1.RRND == 1 is biased and might generate tags from a not uniformly random distribution when certain exclusion masks are being used.
When GCR_EL1.RRND == 0, the IRG instruction generates tags following the ChooseNonExcludedTag() algorithm which is also biased as it was intended for debugging purposes.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs when the following conditions are met:

1. GCR_EL1.RRND == '1'
2. The core executes an IRG instruction.

Implications

If the conditions mentioned above are met, random tag allocation might be biased. Arm does not consider the risk of a bias in the generation of random tags to change the security guarantees and value of MTE, or have any practical security impact.

Workaround

No workaround is expected to be required for this erratum.

3705320
Interrupt signals generated by cores might be deasserted when in retention

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

If a core enters the FULL_RET power mode while one of the interrupt outputs is asserted, the interrupt output might be deasserted without the interrupt having been acknowledged.

Configuration affected

This erratum affects all configurations.

Conditions

The erratum occurs if all of the following conditions apply:

1. One of the following interrupt outputs generated by the core is asserted:
    - **nTBEIRQ**
    - **nPMUIRQ**
    - **nCOMMIRQ**
    - **nVCPUMNTIRQ**
    - **nCOREERRIRQ**
    - **nCOREFAULTIRQ**
2. The core executes a **`WFI`**, **`WFIT`**, **`WFE`**, or **`WFET`** instruction and enters the FULL_RET power mode.

Implications

If the conditions are met, then the asserted interrupt outputs might become deasserted while in the FULL_RET power mode. Once the core exits FULL_RET, the interrupt output will be asserted again. If the interrupt is relied upon to wake the core from the **`WFI`**, **`WFIT`**, **`WFE`**, or **`WFET`** instruction, and no other wake-up events occur, the core might never wake up.

Workaround

No workaround is expected to be required for this erratum.

3729900
Performance might drop for core 1 in a 2-core complex configuration

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2 and r0p3. Fixed in r0p4.

Description

Performance might drop for core 1 in a 2-core complex configuration.

Configurations affected

This erratum affects all configurations with a 2-core complex, and the L2 cache is present (parameter L2_CACHE set to True).

Conditions

The erratum occurs under the following condition:

- Software runs on core 1 in a 2-core complex configuration

Implications

The performance of software running on core 1 might have a noticeable performance drop compared with software running on core 0. The performance drop depends on the software workload. Arm observed a 2.5% performance drop on core 1 for a Specint2k17 run.

Workaround

No workaround is available.

3738908
Consecutive CTI trigger events from the same ELA output might not be sent to the CTI

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

When consecutive *Cross Trigger Interface* (CTI) trigger events from the same *Embedded Logic Analyzer* (ELA) output occur, the core will only send the first one to the CTI. New trigger events from the same ELA output will not be sent until a different CTI input trigger event or a debug event occurs, or the core is reset.

The CTI input trigger events are as follows:

- Cross-halt
- *Performance Monitoring Unit* (PMU) counter overflow
- 4 *Embedded Trace Extension* (ETE) trace unit external outputs
- 2 ELA outputs

The type of debug events are as follows:

- The processor leaves cold reset.
- The processor indicates to the External Debugger that the core has entered or left debug state.

Configurations affected

This erratum only affects configurations that include ELA.

Conditions

The erratum occurs under the following conditions:

- ELA is programmed to drive one of its trigger outputs
- The trigger event occurs twice, without any intervening CTI input trigger events or debug events or reset.

Implications

Triggers are used to perform debug actions such as entering halting debug mode or starting trace. Missing triggers will prevent these actions from happening, without any further implications for future triggers or debug actions.

Workaround

No workaround is expected to be required for this erratum as the impact is limited to debug only, and is specifically limited to ELA, which should not be present in production devices.

3762369
Incorrect ESR_EL1.ISS.EX or ESR_EL2.ISS.EX for Software Step Exceptions

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

After Software-Stepping over a Load exclusive instruction, and under certain micro-architectural conditions, the core might report ESR_EL1.ISS.EX or ESR_EL2.ISS.EX as 0x0 instead of 0x1.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum can occur if all of the following conditions apply:

- Conditions to enter Software-Step state:
  - MDSCR_EL1.SS = 1
  - SPSR_ELx.SS = 1
  - ELR_ELx points to an instruction being stepped
- ERET to enter Software-Step execution state.
- One of the following Load exclusive instructions is executed in Software-Step mode:
  - LDXP
  - LDXR
  - LDXRB
  - LDXRH

- Timing-sensitive micro-architectural conditions occur.

Implications

Under the conditions previously defined, the core might report ESR_EL1.ISS.EX or ESR_EL2.ISS.EX as 0x0 instead of 0x1.

Workaround

No workaround is expected to be required for this erratum.

3777128
CTI trigger events from a core in Standby state might not be sent correctly

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

The debug and trace events which are connected to *Cross Trigger Interface* (CTI) input triggers might not be sent, be delayed, or be sent multiple times while a core is entering Standby state.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs under the following conditions:

1. CTI is enabled by setting CTICONTROL.GLBEN to 1.
2. The core is entering Standby state after executing one of the WFI, WFE, WFIT, or WFET instructions.
3. During the transition to Standby state, one of the following affected trigger events occurs with microarchitectural timing conditions:
    - *Performance Monitoring Unit* (PMU) counter overflow
    - *Embedded Trace Extension* (ETE) trace external output
    - *Embedded Logic Analyzer* (ELA) trigger event

Implications

Under different microarchitectural timing conditions one of the following can occur:

1. A trigger might get delayed until the core exits Standby state
2. In the case of ELA trigger, it might not be observed at all
3. The trigger event might be observed multiple times, which might cause deadlocks

Triggers can be used to request debug actions such as halting mode or starting instruction trace. Missing or delaying triggers mainly affects those debug actions. All of these implications are mitigated by typically low core activity after a Standby instruction, as all trigger events are in response to some core activity.

The microarchitectural conditions for the third implication are particularly rare. The impact of observing the same trigger multiple times in rapid succession on debug actions is low, as the typical delay in responding to a trigger will mean the action is performed only once, and the duplicate triggers will have no effect. However, it is possible that the multiple triggers in flight could cause a deadlock between other types of traffic or power transitions.

Workaround

No workaround is expected to be required for this erratum.

3777132
External debug request while transitioning to emulated off might cause core to deadlock

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1, r0p2, r0p3 and r0p4. Open.

Description

The transition to Emulated off power mode might not complete, and any following requests for power mode transitions will be blocked, when an external debug request is sent to a core that is transitioning to Emulated off power mode. In this state, the core will not wake from standby for any reason or enter halting debug mode.

Configurations affected

This erratum affects all configurations.

Conditions

The erratum occurs if the following sequence of conditions occur:

1. Either of the following is performed to prepare for entry to Emulated off power mode:
   - Dynamic entry to Emulated off power mode is enabled using the PPU_PWPR register in the core *Power Policy Unit* (PPU).
   - The core requests entry to Emulated off power mode by writing to the DBGPRCR_EL1 and IMP_CPUPWRCTLR_EL1 registers.
2. The core finishes preparing for powerdown and executes one of the following Standby instructions:
   - WFI
   - WFE
   - WFIT
   - WFET
3. An external debug request arrives at the core under microarchitectural timing conditions before the transition to Emulated off power mode completes.

Implications

Under the conditions previously defined, there is no way to recover from the deadlocked state.

Workaround

No workaround is expected to be required for this erratum as Emulated off power mode is not expected to be used in production devices.

3817217
Affinity ID info in Complex_RAS.ERRDEVAFF register is not correct in single-core complex configuration

Status

Fault type: Programmer Category C
Fault status: Present in r0p0, r0p1,r0p2,r0p3 and r0p4. Open.

Description

The Complex_RAS.ERRDEVAFF register value is not correct in single-core complex configurations.

Configurations affected

This erratum affects all configurations with a single-core complex.

Conditions

The erratum occurs if the following condition applies:

- Read of Complex RAS node ERRDEVAFF register

Implications

Reading the Complex RAS node ERRDEVAFF register returns the ERRDEVAFF.F0V bit to be 0, indicating ERRDEVAFF.Aff0 is not valid. This is not correct for a single-core complex configuration.

Note, with the deprecation of this register in the Advanced Configuration and Power Interface (ACPI) for the Armv8-A specification, the topology is described in firmware tables instead.

Workaround

There is no workaround required for this erratum.

Proprietary notice

(PRE-1121-V1.0)

Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in the Arm documents.

Product status

All products and Services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

Product completeness status

The information in this document is Final, that is for a developed product.

Product revision status

The rxpy identifier indicates the revision status of the product described in this manual, where:

rx

Identifies the major revision of the product.

py

Identifies the minor revision or modification status of the product.